Semantic Multimedia Modelling & Interpretation for Annotation

Irfan Ullah

Submitted in partial fulfilment of the
Requirements for the degree
of Doctor of Philosophy

School of Engineering and Information Sciences

MIDDLESEX UNIVERSITY

May, 2011

Declaration

I certify that the thesis entitled Semantic Multimedia Modelling and Interpretation for Annotation submitted for the degree of doctor of philosophy is the result of my own work and I have fully cited and referenced all material and results that are not original to this work.

Name ……IRFANULLAH……………………….

[print name]

Signature …………………………

Date………MAY 24TH, 2011……………….

Abstract

"If we knew what it was we were doing, it would not be called research, would it?" Albert Einstein

The emergence of multimedia enabled devices, particularly the incorporation of cameras in mobile phones, and the accelerated revolutions in the low cost storage devices, boosts the multimedia data production rate drastically. Witnessing such an iniquitousness of digital images and videos, the research community has been projecting the issue of its significant utilization and management. Stored in monumental multimedia corpora, digital data need to be retrieved and organized in an intelligent way, leaning on the rich semantics involved. The utilization of these image and video collections demands proficient image and video annotation and retrieval techniques.

Recently, the multimedia research community is progressively veering its emphasis to the personalization of these media. The main impediment in the image and video analysis is the semantic gap, which is the discrepancy among a user's high-level interpretation of an image and the video and the low level computational interpretation of it. Content-based image and video annotation systems are remarkably susceptible to the semantic gap due to their reliance on low-level visual features for delineating semantically rich image and video contents. However, the fact is that the visual similarity is not semantic similarity, so there is a demand to break through this dilemma through an alternative way. The semantic gap can be narrowed by counting high-level and user-generated information in the annotation. High-level descriptions of images and or videos are more proficient of capturing the semantic meaning of multimedia content, but it is not always applicable to collect this information.

It is commonly agreed that the problem of high level semantic annotation of multimedia is still far from being answered. This dissertation puts forward approaches for intelligent multimedia semantic extraction for high level annotation. This dissertation intends to bridge the gap between the visual features and semantics. It proposes a framework for annotation enhancement and refinement for the object/concept annotated images and videos datasets. The entire theme is to first purify the datasets from noisy keyword and then expand the concepts lexically and commonsensical to fill the vocabulary and lexical gap to achieve

high level semantics for the corpus. This dissertation also explored a novel approach for high level semantic (HLS) propagation through the images corpora. The HLS propagation takes the advantages of the semantic intensity (SI), which is the concept dominancy factor in the image and annotation based semantic similarity of the images. As we are aware of the fact that the image is the combination of various concepts and among the list of concepts some of them are more dominant then the other, while semantic similarity of the images are based on the SI and concept semantic similarity among the pair of images. Moreover, the HLS exploits the clustering techniques to group similar images, where a single effort of the human experts to assign high level semantic to a randomly selected image and propagate to other images through clustering.

The investigation has been made on the LabelMe image and LabelMe video dataset. Experiments exhibit that the proposed approaches perform a noticeable improvement towards bridging the semantic gap and reveal that our proposed system outperforms the traditional systems.

Acknowledgement

"No duty is more urgent than that of returning thanks [Schaff and Wace, 2002]."
ST. AMBROSE Bishop of Milan (340 – 397)

I was once told that PhD is a long journey of transformation from a 'novice' to 'professional' researcher. To succeed, an individual can never do it alone; there must be someone who is there for them, providing all the helping hands and supports. I can't agree more and as for my case, I have a lot of people to thank.

I owe my gratitude to all those people who have made this thesis possible. It is a pleasure to thank them all in my humble acknowledgment In the first place, I am heartily thankful to my supervisor Dr. Jonathan Loo (Middlesex University) for helpful discussions, guidance and support throughout my PhD studies. His scientific intuition and rigorous research attitude have enriched me with the in-depth understanding of the subject and the passion of dedicating myself to research, which I will benefit from in the long run. His gracious support in many aspects has enabled me to overcome difficulties and finish my study. I also extend my thanks to my second supervisor Dr. Martin Loomes.

Further, I would like to express my special gratitude to my colleague and best friend Nida Aslam, whose dedication, love, and persistent care has diminished all the difficulties of studying and living in the unfamiliar land far away from home.

I am grateful to my financial supporters, including the Kohat University of Science & Technology, Pakistan. Many thanks also go to my friends shafi Ullah khan, Tahir Naeem who accompanied me during the last three years, making my time in London most enjoyable. Finally, I pay tribute to the constant support of my family, but especially of my parents, Brother, sister and my wife & kids (Mahnoor & Lutfullah) for their untiring efforts and praying. Without them, my whole studies would have been impossible and whose sacrifice, I can never repay. This one is for you!

.

Dedication

I dedicate this thesis to my late Father, (may his soul rest in peace) for their unconditional support and prayers, tireless love and motivation throughout my studies. Everything I have accomplished, I owe to him.

List of associated Publications

1. **Irfan Ullah**, Nida Aslam, Jonathan Loo, RoohUllah "Adding Semantics to the reliable Object Annotated Image Databases " World Conference on Information Technology, 2010.

2. **Irfan Ullah**, Nida Aslam, Jonathan Loo, RoohUllah "A Framework for High Level Semantic Annotation Using Trusted Object Annotated Dataset" IEEE International Symposium on Signal Processing and Information Technology December 15-18, 2010 - Luxor – Egypt.

3. **I.Khan**, N. Aslam, K.K. Loo, "Semantic Annotation Gap: where to put the responsibility", International Journal of Digital Content Technology and its Applications (JDCTA), ISSN: 1975-9339, Vol. 3, No. 1, March 2009.

4. **I.Khan**, N. Aslam, K.K. Loo, "Semantic Multimedia Annotation: Text Analysis", International Journal of Digital Content Technology and its Applications (JDCTA), ISSN: 1975-9339, Vol. 3, No. 2, June 2009.

5. **Irfanullah**, Nida Aslam, Jonathan Loo, Roohullah. "A Framework for Image Annotation Enhancement & Refining Using Knowledge Bases". Springerlink International Journal on Digital Libraries (Submitted Feb 2011).

6. **Irfanullah**, Nida Aslam, Jonathan Loo, Roohullah. "Semantic Space Enhancement and Refinement for Video using Knowledgebases". Elsevier Journal of King Saud University – Computer and Information. (Submitted Feb 2011)

7. **Irfanullah**, Nida Aslam, Jonathan Loo, Roohullah. "Exploiting the Semantic Intensity for High Level Semantic Annotation of Images". Elsevier Journal of Visual Communication and Image Representation (Submitted Feb 2011)

8. **Irfanullah**, Nida Aslam, Jonathan Loo, Roohullah. A survey on exploiting knowledge bases for visual media annotation.

Table of Contents

List of Figures

List of Tables

List of Acronyms

CBIR	Content Based Image Retrieval
HLS	High Level Semantic
EXIF	Exchangeable image file format
GPS	Global Positioning System
XML	Extensible Mark-up Language
RDF	Resource Description Framework
MPEG	Moving Picture Experts Group
COMM	Core Ontology for Multimedia
VIA	Video Image Annotation
LSCOM	Large-Scale Concept Ontology for Multimedia
SI	Semantic Intensity
TBIR	Text-Based Image Retrieval
AIA	Automatic Image Annotation

Chapter 01

Introduction

"Logic will get you from A to B, Imagination will take you everywhere"
Albert Einstein

Annotation is a methodology for adding information to a multimedia document at some level—a word or phrase, paragraph or section or the entire document. This information is called "*metadata*," that is, data about other data. The difference between annotation and other forms of metadata is that an annotation is grounded to a specific point in a multimedia document. For example, one might consider a folder name on a computer as metadata for the files in that folder. So, a folder labelled *"holiday 2010"* might hold files of photographs taken on holiday. The folder name is a form of metadata. But, when an image file is taken out of the folder, it becomes separated from that metadata and thus loses some valuable context. Thus, the data that stays with the image and that describe the entire contents of the image is called annotation. The task of content annotation is to enrich the audio-visual content metadata and data that describes the content. Content analysis can thus be seen as reversing the authoring process, during which an audio-visual material is created based on information about the content to be produced.

Images taken from digital cameras, for example, are rarely annotated by consumers. Images are usually automatically recorded in meaningless alphanumeric filenames. Many people attempt to manage their digital images by annotating them manually, which is very time consuming and often subject to individual interpretation. As a simple solution, the images are archived in file system folders according to their semantics such as an event, a venue and a person of interest. But in reality, users need assistance for finding their way in this overload of digital information. Today's search engines have achieved satisfying quality for textual information, but not for multimedia. The reason is that "*a word is easily identifiable in a text and is usually associated with a concept at the level of human communication. An automatically identifiable feature of an image, such as a colour distribution, does not provide a retrieval system with a concept that is equally useful for user interrogation"* [Ribeiro et al 2001] and is therefore not practical for indexing as is required by search engines. The available, searchable information for multimedia (such as filename or perhaps title, author and file format) is seldom sufficient for achieving pleasing search results. For effective retrieval, the semantic annotation of the still and moving images or visual resources is the central topic of this thesis.

The rest of the chapter is organized as follows. In the next section, we introduce the existing trends in the market in term of multimedia and in broad sense the problem faced by the research community in dealing such type of data. In section 1.2, we focus on the motivation and application of this research, which is further discuss in detail as a research aims and objectives in section 1.3. In section 1.4, the existing problem and challenges are discussed, while section 1.5 covers the research direction. 1.6 focuses on the research contribution of this thesis and finally, the chapter is concluded with an outline of the thesis in section 1.7.

1.1 Introduction

In early 1960, the first computer-based use of multimedia data was developed, which tried to unite the images and text in a document. Subsequently, more and more continuous media, e.g., audio, animations, and video, were incorporated in multimedia systems. Nowadays, most people refer to multimedia as the idea of combining different media sources into one application [Lawrence et al 2004], such as broadcast news video that uses text, images, and audio to describe the progress of news events. Interest in the production and potential of digital data has increased greatly in the past decade, also the storage costs have dropped to the point where user need hours, not minutes, of high-quality video to fill a standard hard disk. Digital data both images and videos are produced by a variety of devices such as digital cameras, camcorder, scanners, co-ordinate measuring machines, airborne radars and digital synthesizers. Digital data can also be created and modified by using multimedia editing software. The comprehensive use of digital technologies causes production of millions of images and videos daily. Adding to this, the growing amount of legitimate content from companies such as Apple Computer, Flickers, YouTube, and Google Video, and the scale of consumers demand for video begins to emerge as shown in the Figure 1.1 the increase in the digital contents and their technology in the past 5 years. However, if all these digital data are not manageable and approachable by general users, they will come to be much less useful in practice. This statement has been reflected in one of the SIGMM grand challenges [Lawrence et al 2004]:

> *"Making capturing, storing, finding and using digital media an everyday occurrence in our computing environment".*

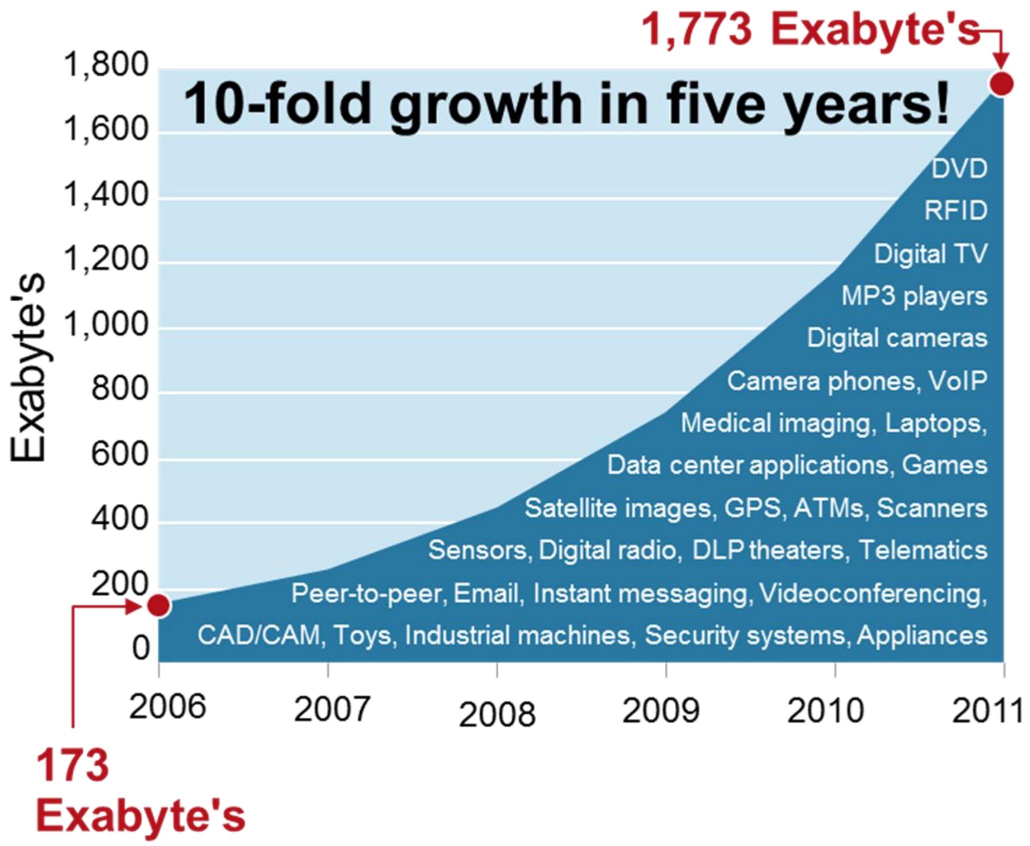


Figure 1.1: Data Production Exponential rate [John et al. 2008] Amount of Digital Information Created and Replicated each year.

The alternative and most appropriate solution is the development and production of metadata which is an additional data, often in textual form, attached to multimedia or other resources for the purpose of describing them and has been identified as a way to compensate limited searchability. Once it has been established for a multimedia, search engines can index the given descriptions in the same way they index textual documents. Thereby, search for multimedia on a higher, conceptual level is enabled. There is a major problem of how to produce metadata for multimedia. Only so-called low-level features like predominant colour or shape can automatically be extracted and then translated into metadata. High-level, conceptual features of multimedia, such as topics of a discussion, story line of a movie, or entire semantic of the image or video cannot be recognized in a reliable way by computers. Those features need yet to be extracted and annotated either by human experts, computer or using hybrid approach of human and computer.

Human beings have the capability to interpret images at various levels, for example, by the colour and texture, objects, proper nouns and emotions. The interpretations can be represented in high-level semantics such as *"sad", "husband"* and *"president"*. The only way a machine is able to interpret images is through examples of visual image feature descriptors or low level image features that represent colour, shape and texture in numerical format. This in turn, introduces an interpretation inconsistency between image descriptors and high-level semantics as shown in the Figure 1.2 and is known as semantic gap [Santini et al 1998, Smeulders et al 2000], which is defined as follows:

> *"The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation"*

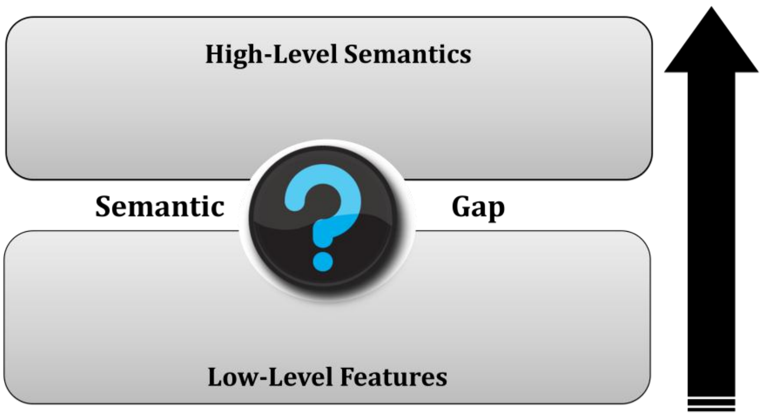


Figure 1.2: Semantic Gap between High-Level-Semantic and Low-Level-Features

This is due to the fact that the visual image feature descriptors extracted from an image cannot (as yet) be automatically translated reliably into high-level semantics [Datta et al 2008]. The broad spectrum of the semantic gap in multimedia is presented in Figure 1.3, the focus of this thesis in broad sense is to bridge the semantic gap between the low-level features and high level semantic concept extractions. This problem is further elaborated in Figure 1.4, which shows the semantic gap hierarchical representation levels from pixels to semantically correct concept extraction. A large semantic gap exist between objects annotation and semantics i.e. how to deduce a high level concepts that what is happening in the image or what is the entire story of the image or multimedia in general and this is the main focus of this dissertation.

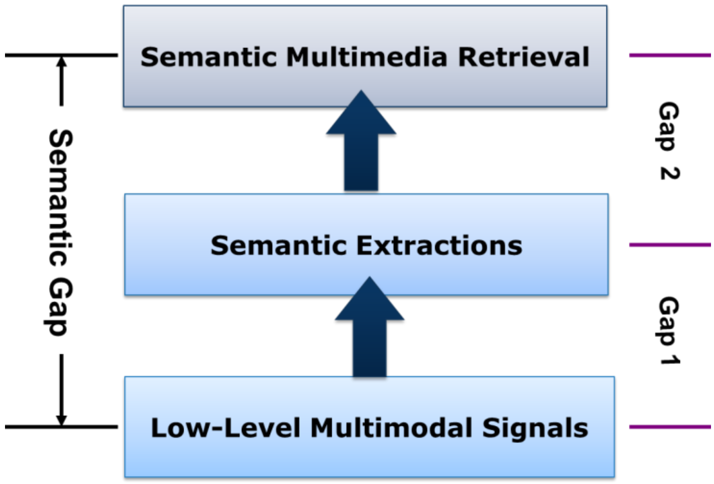


Figure 1.3: Broad picture of the semantic gap in multimedia. Mainly two major semantic gaps exists (1) the gap between low-level features and (2) the semantic gap exists between high level extracted concepts and semantically correct retrieval of the multimedia documents.

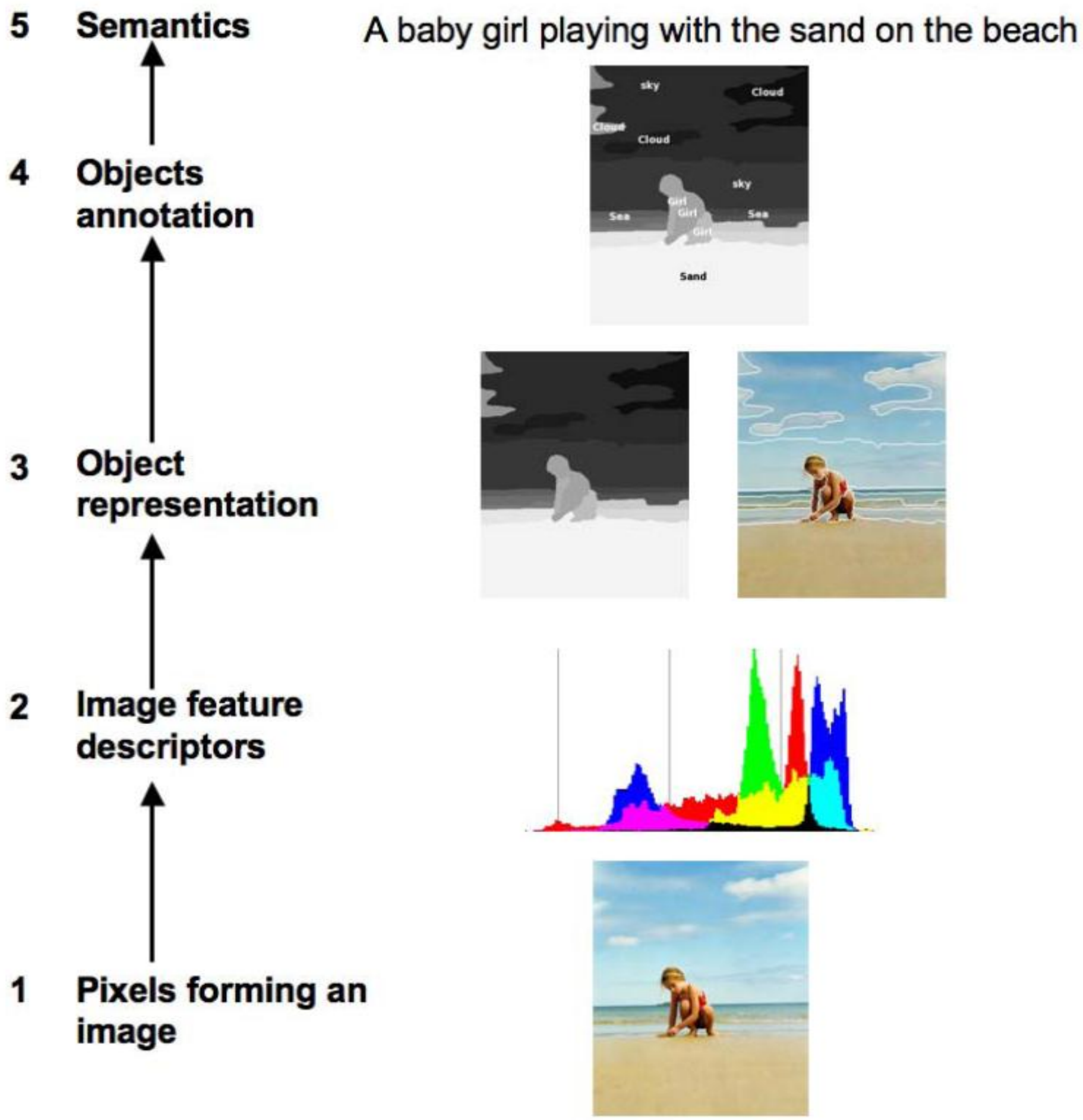


Figure 1.4: [Iskandar 2008] The semantic gap hierarchical representation levels from pixels to semantics, a large semantic gap exist between objects annotation and semantics i.e. how to deduce a high level concepts that what is happening in the image or what is the entire story of the image.

The emphasis of this dissertation is the semantic description, understanding, and modelling of multimedia. The goal is to reduce the semantic gap between the multimedia understanding of the human and the computer by developing a multimedia representation that allows describing them. The work focuses especially on the semantics multimedia interpretation and modelling for annotation. Recently often used in the context of content-based multimedia description, semantics is actually an area in linguistics that deals with the sense and the meaning of language and the question how to deduce the meaning of complex concepts from the meaning of simple concepts. Because of the linguistic background, for us, semantic description implies verbal description, and we thus aim for a description of multimedia based on keywords. The main idea is to extents the simple object annotated datasets of the multimedia by using knowledgebase that not only supports in understanding the multimedia contents semantically but also extends and verify the already extracted concepts.

1.2 Motivation and Application

Research on multimedia annotation is mainly motivated by people's increasing needs for handling large set of multimedia. With the large amount of multimedia data available favoured by cheaper and cheaper digital imaging and digital storage devices, there is an urgent need for an efficient management, indexing and retrieval system. Early image retrieval systems relied on keyword annotation and can be dated back to 1970's as suggested by Chang [Chang et al 1992]. In such approaches, images are first manually annotated with textual keywords. As long as the annotation is accurate and complete, keywords can provide an accurate representation of the semantics of images. However, manually annotating images requires a large amount of human labour, and prone to error as different people can give or inconsistent annotations to the same images. Although it is possible to annotate web images by their associated texts, such as titles, captions, URL's and surrounding texts, these annotations are still very noisy and they are not applicable to non-web images.

To overcome the above difficulties, an alternative scheme, content-based image retrieval (CBIR) was proposed in the early 1990's by Huang et al. [Arnold et al 2000]. In these CBIR systems, various low-level visual features are extracted from a dataset and stored as image index. A query is an image example that is indexed by its features, and retrieved images are ranked with respect to their similarity to this query index. Given that indices are directly derived from the image content, this process requires no semantic labelling. Its advantage over the keyword-based image retrieval is that the feature extraction can be performed automatically and the image's own content is always consistent. However, despite a great deal of work in CBIR, its performance is far from satisfactory due to the semantic gap between visual features and symbolic concepts. That is, images of different semantic content may share some common low-level visual features, whereas images of the same semantic content may be scattered in the feature space, as an example shown in Figure 1.5. Although, today research in the field of low-level features for detecting and recognizing the objects in the images are most mature and there a lot of system available that can do this without human intervention. But in case of high level semantic there is a need to bridge the semantic gap between object annotations to semantic representation of the multimedia.

Figure 1.5: Images that visually looks similar but not semantically similar.

In order to narrow down or bridge this semantic gap, a large number of works has been done on semantic multimedia annotation using with or without knowledgebase. Also a lot of work has been done on the automatic annotation of multimedia, with the aim of allowing annotating with a minimal human assistance. The motivation of this dissertation is to bridge the semantic gap between object annotations and semantically representation of the multimedia by using knowledgebases and device a framework that not only provide semantic representation of the multimedia in specific domain but also work for general multimedia.

The intended audience for this specific research comprises most of the companies that acquire

1) Helps in managing the multimedia data effectively and efficiently.

2) Helps in searching and retrieving the particular piece of information from the large dump of information. It makes media search and retrieval easy.

3) Helps in managing the security data CCTV.
4) Content Owners -- Production companies like BBC, CNN, Geo news etc.
5) TV Service Providers -- Satellite & Cable companies
6) Electronics Manufacturers -- Mobile, DVRs, Digital media players
7) Internet Protocol TV software developers like Microsoft and Virage.
8) Content-Service Providers
9) Content monitoring companies which provide push and pull services
10) Web-Content aggregators
11) Companies that aggregate digital media like Google, Yahoo, YouTube, Flickers etc.
12) Content-repackaging companies
13) Companies that acquire content like sports videos and TV programs and repackage it according to user needs

1.3 Research Aims and Objectives

In general the aim of this work is to investigate promising approaches to extract high level semantic from the multimedia object annotated datasets with the help of knowledgebases, by either utilising or modifying different existing techniques or device a novel framework for the same. The object annotation of a multimedia consists of one or more textual keywords, each describing some specific semantic concept, such as *"sky, sunset, tree, people, beach"*. Despite many efforts by researchers in the last decade, this objective has remained, for the most part, unsolved. Although reasonably successful attempts have been made for some special concepts, such as human faces and people, no satisfactory methods exist that work well with high level semantic concepts in general.

The mainly objective is to focus on exploring the techniques for semantic concept extraction i.e. high level semantic annotation with the help of knowledgebases that can be applied for both images and videos and can be extend to other domain as well by integrating a domain specific knowledgebase. Semantic concepts related to the multimedia are the main requirements to show that the indexing method is feasible; that is to support the search and retrieval with high accuracy.

More specifically, the objectives of the research project are as follows:

1) To solve the problem of high level semantic annotation

   a) To address the issue in semantic annotation and the related work.

   b) To investigate various techniques developed and used for semantic annotation and multimedia datasets indexing.

2) To explore the different knowledgebases and select suitable one that can support the annotation from mid-level to high-level semantics.

3) To formulate a framework for annotation at high level of semantics and develop a system based on this framework.

   a) To develop a suitable algorithm for high level semantic extraction, knowledgebases utilization and indexing.

   b) To design and develop an automatic system that extends the semantic space of the existence annotation.

   c) The existing work can easily be integrated to domain specific by integrating the domain knowledgebase.

   d) To conduct a set of evaluation with different evaluation parameters that can signify the strength of this research proposed.

1.4 The Existing Problems and Challenges

Annotation and retrieval of multimedia data has, without a doubt, received much attention in the last decade, both from a research and a commercial viewpoint. The amount of data that exists and continues to be created is unfathomable, to the point where the data starts to lose its intrinsic value. What good is data if the valid information and meaning that it contains cannot be extracted? A digital camera, for example, allows a person to save thousands of pictures on a hard drive while a digital camcorder eats gigabytes of space to store hours upon hours of footage. If that was not enough, digital audio compression has turned computers into super jukeboxes. As exciting as these applications are, it is becoming increasingly evident that maintaining all this digital data is becoming a daunting task. Thousands of pictures on a hard drive become useless if we cannot find a specific image or a group of images in which we are interested. If we cannot find scenes of interest in video footage, it too loses its value as do music files if specific songs or music genres cannot be found. This problem is reflected in Figure 1.6 in the Longtail scenario, where a few gigabytes of images get search hits from most of the search engines, while thousand gigabytes of images get few search hits and millions gigabytes of images are either gets during achieving process or from the owner who knows the exact name or related information of the images. Thus we are beginning to be more concerned with what to do with digital data rather than how to create it.

These new consumer demands have bolstered research that aims to use computers and machine understanding to analyze digital data to extract useful meaning. This has given birth to the flourishing area of multimedia annotation.

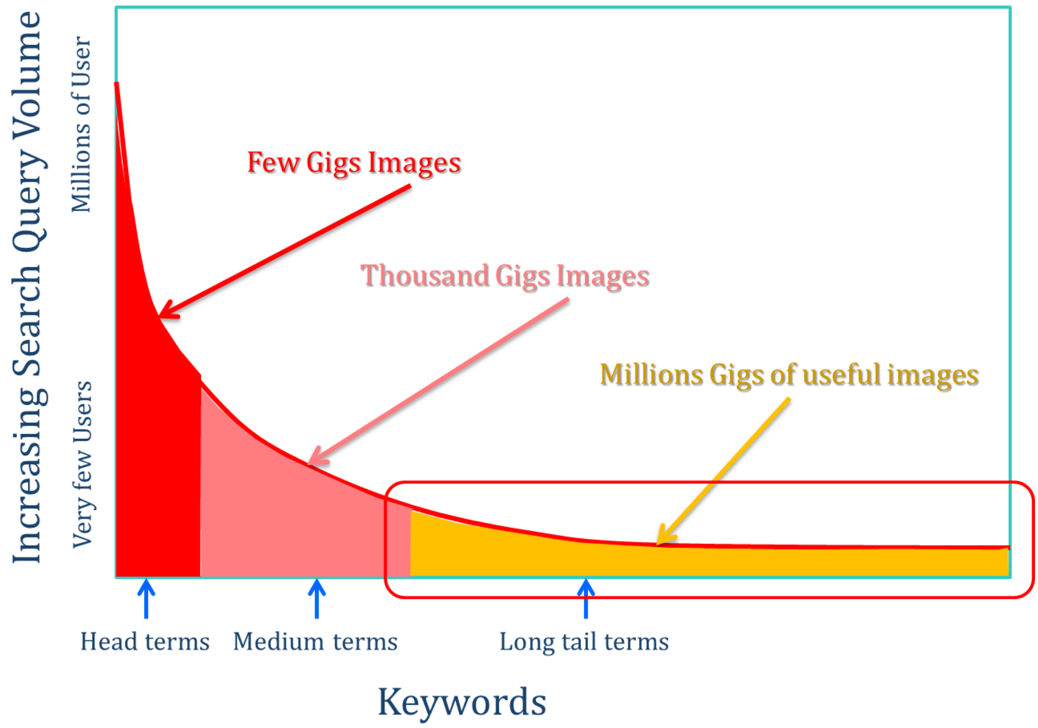


Figure 1.6: Longtail problem for multimedia data, the main challenge is how to make available Giga-byte of multimedia document at Head Term position.

The application of signal processing and computer vision methodologies to images, video, and audio to extract information has initially been done at a low level (e.g., find specific colors or textures in an image). Such features, however, do not contain any *meaning* of the underlying content. For example, it would prove quite attractive to consumers if they could retrieve all the pictures that contain the Eiffel Tower from their large personal image database or if they could record a soccer game and automatically play back only the highlights. Further applications could automatically sort their digital audio collection into different genres or play back only the action scenes of a DVD movie. In other words, there exists more appeal and versatility in being able to retrieve multimedia data based on *semantic* meaning: high-level concepts that relate to language and logic. The ubiquitous nature of multimedia data, the push for manufacturers to create new products and applications, and the improvement in accessibility and speed of computing devices has caused an increase in research and development in the area of semantic annotation of multimedia.

1.5 Research Directions

Many advances have been made in various aspects of multimedia annotation, including visual content extraction, multi-dimension indexing and system design. However, we are still far away from a complete solution for semantic multimedia annotation because there are still many research directions and issues that need to be solved. These include:

1.5.1 High-Level Semantic Concepts and Low-Level Visual Features

Human tends to use high-level semantic concepts in daily life. However, what current computer vision techniques can automatically extract from image are mostly low level features. We have seen that in some constrained applications, such as human face and fingerprint, it is possible to link low level features to high-level semantics (face or finger print). In a general setting, however, the low-level features do not have direct links to high level semantics. To narrow down this semantic gap, some off-line processing can be performed to extract some level of semantics by using either supervised/unsupervised techniques or using some external knowledgebases/ontologies that fill the gap between mid-level and high-level semantics because the knowledgebases/ontologies provide inter-relationship between objects and upsurge the exactitude in the semantics at high level.

1.5.2 Variation of Objects in the Multimedia

There is a large amount of variation in the object annotation of each specific concept. It is worth noting that multimedia object annotation can be thought of as being even more challenging than object recognition because of the diversity of concepts existing in the vocabulary. All the challenges existing in object recognition also exist in annotation. These include viewpoint change of object, background clutter, intra-class variation, occlusion and illumination changes.

1.5.3 Concept Gap and Vocabulary Size

There are large semantic gaps. Some concepts, such as *"yellow", "sport"* and *"car",* are not traditional object concepts, while these properties are mostly annotated by the human experts and their visual appearance is not well-defined or sketched. Learning a direct link from these concepts to semantics is challenging if not possible. Similarly the size of tag vocabulary can be having varied size. The aim of semantic multimedia annotation is to describe the entire semantics of still and/or moving images using a set of textual keywords. Since any word in any language is qualified to be annotated to an image, the possible vocabulary size is nearly unlimited. This greatly increases the complexity of the annotation systems.

1.5.4 Diverse Nature of the Bench Mark Datasets

The availability of datasets and their annotation standard is another core challenge. The datasets like Coral, TRECVID, LabelMe are developed by keeping different aspects of the annotations in mind. This increase the complexity in firming a flexible system for all types of datasets.

1.5.5 Semantic Reasoning Tools

There are a few semantic reasoning knowledgebases available for annotation. A successful reasoner system for semantic annotation relies on the nodes representing concepts and their inter relationship present in the knowledgebases. However, it is hard to take advantage from more than one knowledgebase. The difficulties lie not only in the interpretation, but also different knowledgebases provide interfacing API for different tools, it’s rigid to implement them on one platform.

1.6 Proposed Research Contribution

In the light of the above mentioned problems, we propose a semantic multimedia modelling and interpretation framework that can offers a semantic accuracy in terms of annotation at high level. The main aim of this dissertation is to propose a novel framework for annotation of the multimedia data semantically. It is in this scope that we try to solve one of the most challenging issues of the semantic multimedia annotation i.e. the *semantic gap*. The research contribution are layout in the Figure 1.7 that address two main elements: Lexically and Conceptually Annotation Enhancement and Refinement for the images datasets, High Level Semantic propagation using Semantic Intensity based images clustering technique, while we have extends these approaches for video as well as a third element of the research contribution. Most of the previous work emphasises on low-level primitive features of the multimedia. With this approach we try to investigate a way to explore what is the possible way that can enlarge the semantic space of the images and videos by utilizing the existing annotation sets. We have substantially reduced the semantic gap and achieve a noticeable improvement retrieval degree, concept diversity and enrichment ratio.

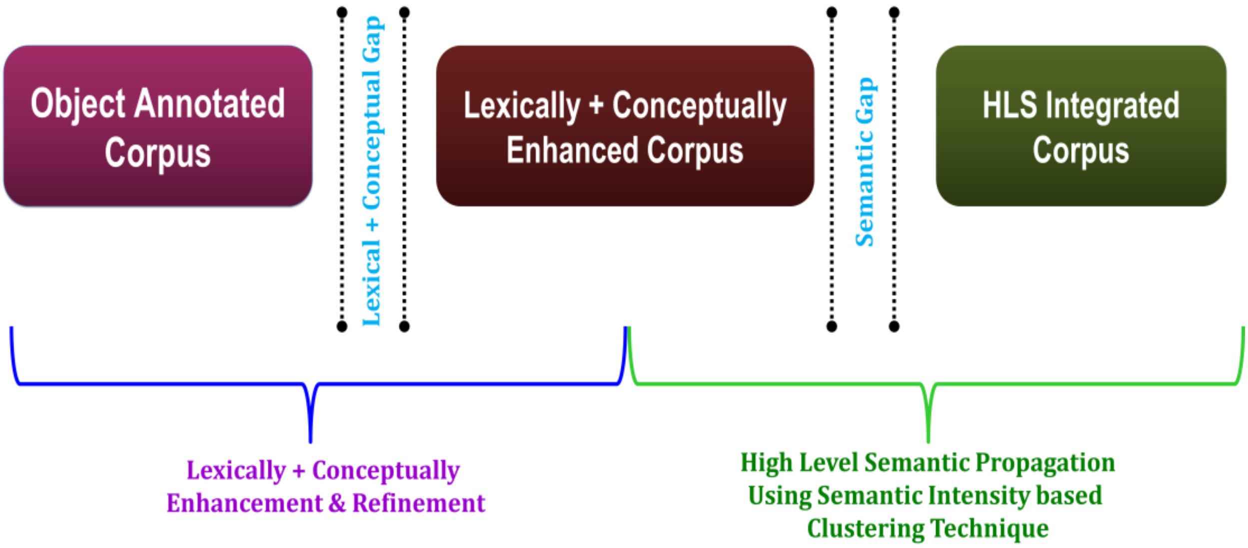


Figure 1.7: Proposed Research Contribution, where Lexical gap, Conceptual gap and semantic gap are tackle as a research contribution.

1.6.1 A Framework for Annotation Expansion and Refinement for Images Dataset

Semantic annotation has become the very important and active research area in the multimedia community. Semantically enriched multimedia information is crucial for equipping the kind of multimedia search potentials that professional searchers need, while on the other side the expansion growth of multimedia (images and video) data online has the potential to encourage more erudite and vigorous models and algorithms to systematize, index, retrieve multimedia and the like corpus. On the contrary, inclusively how much data can be hitched and systematized remains a critical problem, also the semantic interpretation of multimedia is obsolete without some mechanism for understanding semantic content that is not explicitly available. However, Manual annotation is the exclusive source to overwhelming this, which is not only time consuming and costly but also lacks semantic enrichment in terms of concept diversity and concept enrichability as well.

We have proposed semantically enhanced information extraction model that enhances the tagged concept lexically and commonsensically by using the WordNet and ConceptNet by increasing the semantic space for each of the image in the corpus. By doing this a lot of noises, redundant and unusual keywords are generated, which are then filtered out by applying various techniques like semantic similarity, stopwords and words unification.

1.6.2 High Level Semantic Propagation

Multimedia annotation data plays an important role in the future annotation-driven multimedia system. The basic intention of proposed High Level Semantic Propagation is to investigate a mechanism for the ease of manual annotation to a large pool of objects annotated images datasets, where images are clustered based on the annotation and the concept intensity and assigning high level semantic description to them. The research contribution under this head intent to equip the high level semantic annotation for images, and consequently, contributes to 1) calculating concept intensity of each concept in the annotation set of the image depicting the dominancy factor, (2) image similarity on the bases on metadata tag with the images, and (3) image classification and categorization on the basis of their image similarity while high level semantics are then propagate through the image corpus with their calculated similarity values.

1.6.3 Annotation Enhancement and Refinement for Video corpus

Semantic annotation for video is a key to semantic-level video browsing, search and navigation. The research on this topic evolved through three models. The first model applies the binary classification approaches to identify each individual concept in a concept set. It accomplished only limited success, as it did not exhibit the inherent correlation between concepts, e.g., urban and building. The second model added a second step on top of the individual-concept detectors to amalgamate multiple concepts. However, its performance diverges since the errors aroused in the first detection step can propagate to the second fusion step and therefore, degrade the overall performance. The third paradigm focuses on the ontological approach, where a visual knowledge is used to detect concepts and the relationship among them provide an opportunity to inference semantically. As this method depends on rules that are created by domain experts and is suitable for specific domain and is suffer from experts personal knowledge as well. To address the above issues, we propose a forth paradigm which is the extension of the third paradigm from domain specific to general and is based on the concept of text mining approaches where the raw annotated structure of the video are expanded lexically and commonsensically through knowledgebases (i.e. WordNet, ConceptNet). We compare the performance between our proposed approach with the base line ground truth on the widely used LabelMe videos. We report superior performance from the proposed approach.

A detailed discussion on all these contribution has found in the forth coming chapter 3, 4 and 5.

1.7 Organization of the Thesis

The thesis is organized in the following manner.

In Chapter 2, an extensive discussion on the up-to-date achievements concerning the components of Image and video annotation is provided. The main aim of this chapter is to survey the state-of-the-art in the respective field. This includes general images annotation overview along with the overview of the video annotation. The discussion is leading from fundamental concepts to the high-level of semantics, starting from the annotation, its characteristics, standard for multimedia annotation and its type are discussed and then a comprehensive discussion on the multimedia annotation is presented. The discussion about the multimedia annotation is sub-sectioned into three categories i.e. manual, automatic and semi-automatic annotation. Adding to this the temporal based annotation for video is discussed separately. Moreover, comprehensive survey on ontology and knowledgebases based annotation for images and videos are presented. Finally the evaluation measures are discussed and the chapter concluded with summary.

In Chapter 3, a proposed framework for the annotation enhancement and refinement using object annotated datasets is presented. The chapter also explored the recent work in the area of multimedia annotation along with their pros and cons. The efficiency of the proposed system is tested in terms concept diversity, enrichment ratio and retrieval degree. The experiments were performed on open source image dataset (LabelMe) to prove the semantic accuracy of the proposed system.

In Chapter 4, a semiautomatic way for the high level semantic propagation through the images corpus is presented. The work is support by the brief discussion on the state-of-the-art in the respective area. While a new term Semantic Intensity (SI) which depicts the concept dominancy in the image were introduced. The experimental work is performed on the enhanced version from the previous work of the LabelMe datasets, which is a trusted object annotated datasets and a noticeable improvement is achieved in the term of precision and recall.

In Chapter 5, the proposed framework for the annotation enhancement and refinement is extended to video domain. The chapter covers video structure and their representation for the annotation followed by the state-of-the-art of for video annotation and refinement. At the evaluation, the proposed work is evaluated on the LabelMe videos datasets. Results reported to verify the effectiveness of the proposed model.

Finally, in chapter 6 we conclude with a summary of achievements and the future work are discussed. Chapter 6 is followed by appendices and references.

The appendices contain the source code and implementation description of the proposed contributions.

It is to be noted that all the main chapters are presented with a self-contained set of introduction, main concepts, experimental results, and conclusion.

Chapter 02

Fundamental Concept & Literature Review

"Never express yourself more clearly than you are able to think"
Niels Henrik David Bohr

The technological revolution and achievement at present in the field of multimedia was a fantasy a few decades ago. With the advent of wide variety of multimedia enable and/or capturing devices allow an opportunity to anyone to act like a professional and capture photos or even record the event. On other side, the day by day decreases in the cost of the storage devices provide an opportunity to everyone to store photos or recorded events for later use. These progressions persist at an incredible velocity for a commercial purpose multimedia production and consumption as well. The TV and news broadcast channels, social media application like Facebook or video content provider like YouTube, Dailymotion and the like fueling this on daily basis. All these advances yet bought up with a new demand of effective multimedia data management and retrieval.

Today, the retrieval system has achieved the users need for the textual data but for the multimedia data like images and videos it's still at the infancy stage. The reason is that *"a word is easily identifiable in a text and is usually associated with a concept at the level of human communication. An automatically identifiable feature of an image, such as a color distribution, does not provide a retrieval system with a concept that is equally useful for user interrogation"* [Ribeiro et al. 2001] and is therefore, not practical for indexing as is required by search engines. There is a strong need to establish the metadata at an intelligent way that not only describe the image properties, but depicts the entire content of the multimedia as well.

In the past, metadata was often neglected and treated as a second-class citizen. However, once the computer era emerged and people started using computers to store their data, the need for techniques to retrieve these data from computers was established. Since then the metadata concept has evolved in the computer science paradigm, starting from the simple file systems (file names and types) in the early 60s, then database management systems (to describe database fields) in the early 70s, until the 21st century with the advent of the concept of metadata warehouses [Arun, 2004]. Metadata is more important for files in the Web or on a computer which is more abstract and need to be opened to reveal their contents [Milstead et al 1999]. This is especially true for multimedia files.

Multimedia objects/files are data. Basically, there are two ways to represent them. *"The representations of data closer to the sensor level are commonly called low level, and the symbolic levels, high-level."* [Jain, 1994], Features of audiovisual content follow this classification. Low-level features like for instance, hue, saturation and brightness for visual or energy and volume for audio information can be derived automatically from content. High-level features describe the content conceptually on a higher abstraction level and capture the content's semantics. [Mojsilovic, 2001] confirm that *"High-level semantic concepts play a large role in the way we perceive images …"* Also, *"Users typically do not think in terms of low-level features, i.e., user queries are typically semantic (e.g., "show me a sunset image") and not low-level (e.g., "show me a predominantly red and orange image")"* [Vailaya et al., 2001] when querying for multimedia.

Therefore, it is desirable to facilitate retrieval of multimedia based on semantic descriptions rather than on low-level features [Lindley et al, 1998; Mojsilovic et al, 2002; Zhou et al, 2000; Martinez et al, 2000]. The problem is *"that only low-level features (as opposed to higher level features such as objects and their inter-relationships) can be reliably extracted from images [and videos]. For example, color histograms are easily extracted from color images, but the presence of sky, trees, buildings, people, etc., cannot be reliably detected."* [Vailaya et al., 2001]

In a nutshell, metadata constitutes an appealing way to store semantic descriptions and provides a number of *"attractive potential uses: semantic searching, indexing, retrieval and filtering of multimedia databases; image understanding for intelligent vision and surveillance; and conversion between media (speech to text etc.)"* [Page et al., 2001] in the multimedia area. Furthermore, the concept has been tried and refined since it has first been used in a library for books; in case of multimedia the same concept is called *Annotation*, which is data about multimedia [images and/or video].

This chapter reviews basic concepts and relevant literature on metadata, annotation and their techniques for representation of the multimedia semantically. Semantic annotation consists on representing objects, concepts and events inside the multimedia. In section 2.1, we present the fundamental and related concepts with the annotation. Adding to this, the standards for the multimedia annotation and its type are the part of this section is discussed in section 2.2 and 2.3 respectively. The method of annotation of multimedia are discussed in the section 2.4, where the annotation process are further sub-sectioned into manual, automatic and semi-automatic, while discussion about the video temporal annotation is covered in section 2.5. The ontological and knowledgebases integration for multimedia annotation are covered in section 2.6, while the section 2.7 is focus on the refining scheme for multimedia annotation. The evaluation measures are discussed in section 2.8, while the chapter summary is presented in section 2.9.

2.1 Fundamental Concepts

In this section, we will discuss some of the fundamental concepts related to our research starting from the basic to a higher level.

2.1.1 Characteristics of Multimedia for Annotations

Digital media types can be divided based on the modality they stimulate. There are two distinctive classes of media types based on this division: temporal and static media. The characteristic of temporal media is the time dimension, which static media do not possess. Examples of static types of media are images and graphics. Temporal media can be audio, animations, plain video and audiovisual presentations (e.g. movies). Multimedia is a special type of data, which refers to a collection of media types used together. In this context, we relate to multimedia representations and multimedia objects, referring to a multimedia data to which a specific meaning has been added. Annotating temporal media varies from annotating static media.

In case of static media, objects can be decomposed into smaller entities which characterize them. These derived characteristics are called *features* and can be described through *annotations.* Media content features shape the document, and define the modalities they activate. Features come in many forms, and they are usually divided between *high-level* and *low-level* features. *Low-level* features include data patterns and statistics of media content and depend strongly on the content type. *Low-level* feature extraction can be done computationally by automated processes. From images, we can extract statistics on the pixel values, creating color histograms that can be used to classify images. Videos are sequences of images, thus they will share common features. Furthermore with video we can automatically classify the image sequences using also the time dimension. *High-level* features bear more meaningful information than the *low-level* ones. From a color histogram, it is hard to derive meaningful information on the image; for example a green image may indicate to a *forest landscape*, or to a *golf course*. *High level* features represent high level concepts that are meaningful only to humans. The gap between high and low level representations is called the *semantic gap*. Deriving meaningful concepts from *low-level* features of non-speech audio and video in general level is not possible, but focusing to a specific application domain improves possibilities to succeed [Ranguelova, et al 2007]. In some occasions it is difficult to label a feature as *low-* or *high-level.* Frequently in complex classification the term *mid-level* feature is used. The derivation of the characteristics of media objects is called *feature extraction* [Ranguelova, et al 2007]. This process of feature extraction forms a basis for making annotations.

A temporal media can be thought as a sequence of static media objects, thus one could think that its annotation would involve annotating each media object of the sequence one by one. Fortunately, this is not the usual case: changes in the *high-level* features of the content are relatively slow and thus making annotating necessary for only certain events of interest. With *low-level* features, annotating can be performed automatically, thus annotating each object in the sequence is not a problem. There are two main methodologies to temporal media annotation scheme: *stratified* and *segmented*. Segmented is the simplest traditional way of doing it: the idea is to partition the media object into consecutive temporal segments and describe each segment. Commonly, this scheme has been extended to permit grouping kindred segments together, producing a hierarchical multilevel segmentation. Traditional structure of scenes and shots corresponds well to this kind of segmentation. Stratification is a context-based approach to modeling video content. It permits any subsequence of video frames to be modeled as rich multi-layered descriptions that can be easily parsed to support a wide range of applications [Chua et al. 2002]. Ultimately the annotations are organized to form a data structure (i.e. *index*), which is also referred to as *indexing*.

2.1.2 Multimedia Annotation

People seem to use very little time to annotate their personal images. How many amateur photographers are determined enough and have enough time and energy to go through developed pictures, and put them into albums, instead of just sticking the pictures in a shoebox? How many people go through their digital photos and give each one a unique file name in an appropriate directory instead of leaving them in the default directory created by the camera software? Not many [Brown et al 2001]. As a result, more and more people have thousands of digital photos with little or no organization, and they are resigned to gaining no more benefit or enjoyment from them than the photos stored in overfilled shoeboxes around the house. Well-performed annotation has the power to transform this almost random collection of images into a powerful, searchable and rich record of events in people's lives [Jack. et al 2005].

There are two types of information related with a multimedia, which can be either image or video: Structured information about the object, called its metadata, and information contained within the object, called its visual features. Metadata is information connected to the object and can consist of digits and letters that are also referred to as text. It can also consist of sounds sketches or drawings. Visual features are usually automatically extracted from the image. These features are usually size, color, shapes and sketches [Gupta et al 1997].

2.1.3 What is Semantic Annotation of Multimedia?

The term semantic annotation refers to the process of generating a linguistic or natural language description of a given multimedia objects or attaching a textual description to it, i.e. the goal of semantic annotation of still or moving images is to assign semantically meaningful information to images. Text is the most common and relevant way of annotation [Jack. et al 2005]. It provides a description of an image in terms of places, people, events and objects. Multimedia semantic annotation is a part of high-level vision – *"...the highest processing level in computer vision"* according to [Sonka, et al 1999]. Semantic interpretation of an image provides answers to questions such as: *What objects are present in the scene? What location does the image depict? What is happening, what event does it depict? "*

There subsist different levels of comprehension in the hierarchy of semantic annotation: the lowest level is the level of objects. Further up, in order of complexity, understanding entails understanding of the relationships between the objects in the scene (spatial and otherwise). Understanding and interpreting the frame of mind and atmosphere the imaged scene conveys is the most complex task and comes at the very top of the image understanding hierarchy [Levine, et al 1985]. Semantic feature extraction from a video can be done automatically in restricted domains, while broad domain semantic content requires manual annotations. Some features can be extracted with use of knowledge on conventional methods to build scenes. But to gain reliable feature extraction of higher-level features manual annotations are needed.

Moreover, time is an important factor when it comes to image annotation. As times goes by, humans forget what the image is about. This specially applies for images that are hard to identify without having other images of the same context to compare it against. This is also a strong argument for annotation and also a strong argument for doing it right away.

2.1.4 Is Semantic Annotation of Multimedia Feasible?

The research thus far in the area of semantic annotation of multimedia in broad topic still or moving images collections has shown that low-level features on their own do not have sufficient power to bridge the semantic gap between the high-level semantic concepts that humans communicate in, and content-based image description. The potential for filling that void may lie in using other contextual information that may be available. As the availability of lexical and conceptual knowledgebases like WordNet, ConceptNet, CYC, Yago ontology and many domain specific ontologies assist the process from simple object base annotation to semantic annotation. Also capture devices become more powerful, more and more information is recorded at capture time [Ebrahimi, et al 2004]. For instance, the GPS information accompanying a digital photo easily answers the location-question. Dates and times, along with the location information can facilitate an automatic annotation of a photo with semantic labels with respect to the season (winter, spring, summer, autumn) and time of the day (dawn, morning, midday, dusk, night). Likewise, the EXIF's scene brightness tag could help determine whether the photo was taken indoor or outdoor. All this, in turn, could possibly assist other classification and annotation tasks by way of refining their results.

In conclusion, the integration of knowledgebases and raw annotation extracted from image content is likely to offer an improved solution to the task of semantic understanding of the multimedia objects because the knowledgebases has the inter-concept relations and that help not only in depicting the hiding concepts but also provide an opportunity to understand the multimedia with a small number of concepts. Some of the challenges rest in identifying supplementary sources of information as well as finding smart and efficient ways of combining such diverse information. The work described in this thesis explores some of these challenges.

2.2 Metadata of Multimedia Objects

Multimedia (image/video) own text and visual while video have one more modal i.e. audio stream, multimedia documents can be enriched with additional data, the so-called metadata. According to [Blanken et al. 2007], there are various types of metadata, he categories the metadata into three sections (1) A description of the multimedia document, (2) textual annotation and (3) semantic annotations.

2.2.1 Descriptive Data

Descriptive data provides valuable information about the multimedia document. Examples are the creation date, document format, while for video director or editor, length of the video and so on. A standard format for descriptive data is called Dublin Core. It is a list of data elements designed to describe resources of any kind. Descriptive metadata can be very useful when documents within the video collection shall be filtered based on certain document facets. Think, for example, of a user who desires to retrieve all video documents that have been created within the last month, or all videos from one particular director.

2.2.2 Text Annotations

Text annotations are textual descriptions of the content of multimedia documents. Text annotation is often in the form of plain text to describe the entire scene of the multimedia in natural language. This process is mostly feasible in manual annotation, where human expert express the multimedia content in natural language. More recent state-of-the-art online systems, such as YouTube and Dailymotion, rely on using annotations provided by users to provide descriptions of videos. However, comparatively there are often users who have very unusual perceptions about the same video and annotate that video differently. his can result in synonymy, polysemy and homonymy, which makes it difficult for other users to retrieve the same video. The similar problem is facing by LabelMe online annotation tool, where user can sketch the object in the images and tag them. It has also been found that users are reluctant to provide an abundance of annotations unless there is some benefit to the user [Halvey et al, 2007]. [Van Zwol et al. 2008] approach this problem by transferring video annotation into an online gaming scenario by taking idea from the ESP game that perform the same approach for images.

Considering that textual annotations can be a worthwhile source for IR systems aiming to retrieve the multimedia documents, various approaches have been studied to automatically determine textual annotations. But due to the Semantic Gap problem automatically annotating video / images is a non-trivial problem. A survey of state-of-the- art approaches is given by [Magalh˜aes, et al. 2006]. More recent examples include [Stathopoulos et al., 2009; Llorente et al., 2009; Llorente et al., 2008; Qi et al., 2007; Wang et al., 2007b].

2.2.3 Semantic Annotations

Another type of annotations is semantic annotations. The idea is here to identify concepts and define their relationship between each other. Concepts can hence set the content of multimedia documents into a semantic context. This is especially useful for semantic retrieval approaches. The MPEG-7 standard allows for describing multimedia documents and their semantic descriptions. Promising extensions include COMM (Core Ontology for Multimedia), an ontology introduced by [Arndt et al. 2007]. Ontologies are *"content specific agreements"* on vocabulary usage and sharing of knowledge [Gruber, 1995]. Other metadata models include [Durand et al. 2005; Tsinaraki et al. 2005; Bertini et al. 2007], who aim to enrich interactive television broadcast data with additional information by combining existing standards. All approaches build hence upon similar ideas.

Semantic annotations can either be derived from textual annotations or from the image/video low-level features, i.e. by identifying high-level concepts. [Magalh˜aes, et al. 2006] provide a survey on state-of-the-art methodologies to create semantic annotations for multimedia content. They distinguish between three semantic annotation types: (1) hierarchical models, (2) network models and (3) knowledge-based models. Hierarchical models aim to identify hierarchical relations or interdependencies between elements in an image or key frame. Examples include [Barnard and Forsyth, 2001]. Network models aim to infer concepts given the existence of other concepts. Surveyed approaches are [Kumar, et al. 2003; He et al., 2004]. The third approach, knowledge-based models relies on prior knowledge to infer the existence of concepts. [B¨urger et al. 2005], for example, enrich news video data with a thesaurus of geographic names. Therefore, they determine location names within the news reports transcripts and map these with their thesaurus. Further, they identify thematic categories by mapping terms in the transcript with a controlled vocabulary. A similar approach is introduced by [Neo et al. 2006], who use the WordNet lexical database [Fellbaum, 1998] to semantically enrich news video transcripts. Even though their approaches allow linking of related news videos, the main problem of their approaches is text ambiguity. Other examples include [Tansley, 2000; Simou et al., 2005].

2.3 Standard for Annotation to Describe Multimedia

Applying standard technology means reusing expert knowledge, increasing interoperability and saving development costs. Descriptive annotation too can gain substantial benefit from standardization: *"The association of standardized descriptive metadata with networked objects has the potential for substantially improving resource discovery capabilities by enabling field-based (e.g., author, title) searches, permitting indexing of non-textual objects, and allowing access to the surrogate content that is distinct from access to the content of the resource itself."* [Weibel & Lagoze, 1997].

Numerous annotation standards for the (semantic) description of digital resources have been conceived. Many of them can be used to describe multimedia objects. Below, four annotation standards are introduced in chronological order. Each of the standards is measured against the ability to capture the parts of the data model dealing with the description of multimedia objects. Finally, the chosen standard is presented and justified. The oldest and most simple standard is the Dublin Core element set.

2.3.1 Dublin Core

The Dublin Core Metadata Initiative (DCMI) was initiated in 1995. The goal of the group was to make it easier to find resources in the Internet and to advocate the use of interoperable metadata standards. The resulting Dublin Core metadata standard can be used to supply additional information for documents to be used in web-based search and indexing.

The standard targets documents on the Internet. However, it can also be used for other resources, depending on *"how closely their metadata resembles typical document metadata and also what purpose the metadata is intended to serve"* [Hillman, 2001]. Dublin Core's development has been aligned to four design principles:

i. Simplicity
ii. Semantic interoperability between different domains and disciplines
iii. Development in an international effort
iv. Extensibility

The Dublin Core element set was chosen and designed by *"professionals from librarianship, computer science, text encoding, the museum community, and other related fields of scholarship"* [Hillman, 2001]. It consists of 15 elements: title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage and rights. They are listed and described in DCMI (1999). The chosen set resembles typical library cards. One can see them as the *"least common denominator"* of document metadata. Each of the description elements is optional and may be repeated in a metadata record. Dublin Core can be written in several different syntaxes, including generic form, HTML, RDF and XML (see below).

For the description of multimedia objects, Dublin Core is too generic to be useful on its own. It needs to be applied in concert with a more *"powerful"* partner, such as XML [Bray et al., 2000].

2.3.2 XML

XML can be seen as a consequence of the success of the Internet, which made the limitations of HTML evident. HTML is about spatial and styling layout for human consumption. Computers cannot deduce the meaning of content of web pages written in

HTML. Therefore, there was a need for a mark-up language beyond HTML sufficing the following requirements:

- Mark-up for layout and content
- Readable and meaningful for humans and machines
- Flexible and extensible

The metalanguage SGML addressed these issues but was too complicated and not very suitable for the Internet (Geroimenko, et al 2002). Thus, in 1998, XML (eXtensible Markup Language) was conceived as a simpler version of SGML. Basically, XML is plain text with inherent structure. The structure stems from tags. It can be defined by means of document type definitions or Schema files. The tags carry "meta" information about their content. In that way, the meaning the text conveys is increased. Other benefits of XML include:

- XML is an open, vendor independent standard
- XML is plain text, therefore platform-independent.
- XML separates content from its presentation. Different presentation formats can be generated from one and the same source.
- XML contains self-describing information. The tags give hints about the role their content is playing.
- XML is Web-friendly and data-oriented and facilitates integration of data from legacy systems, documents and databases.

XML is a metalanguage. That means that it can be used to define mark-up language customized to particular circumstances. This is very powerful, but has unwanted side effects. According to [Page et al. 2001], *"[the] whole point of metadata is to aid the understanding of other data, so there must be a way to decode the metadata into useful information or it becomes as useless as the data it is augmenting."* If everyone indeed defines a separate language, interoperability and understanding between different organizations will go to zero. The result is a "Tower of Babel" scenario [Geroimenko, et al 2002].

To overcome the problem, developers can write applications translating between XML languages or agree on standards. Many standards have been defined with XML, for instance VoiceXML for mark-up of audio input and output through the telephone, XML Schema, the graphic standards SVG (Scalable Vector Graphics, [Ferraiolo, et al, 2003]) and X3D (Web 3D consortium, 2003) or SMIL [Ayers et al., 2001], a standard for synchronizing different media.

Regarding the semantic description of multimedia objects, XML endues both required power and expressiveness. It is platform independent, is easily transmitted over the Internet and fulfills the criteria discussed before.

2.3.3 RDF

RDF is *"a data model and XML serialization syntax for describing resources* both *on and off the Web."* [Dornfest, et al 2001] It has been developed by the W3C to overcome the problem of incompatible standards for metadata syntax and schema definition languages. RDF targets resource description, site-maps, content rating, electronic commerce, collaborative services and privacy references and is based on web technologies. The main design goal is metadata interoperability. A welcome bonus is machine readability [Ianella, 1998].

Whatever can be labelled with a URI is a resource that can be described by RDF. URI is short for Uniform Resource Identifier and means *"a compact string of characters for identifying an abstract or physical resource."* [Berners-Lee et al., 1998]. A URI identifies a labelled resource unambiguously. Thus, mix-ups are avoided. Each resource is further described by properties. These too have attached URIs. This means they are resources and can be described by RDF.

Definition of properties is decentralized and everyone has the possibility to define new properties. Of course, this is not the intention. Instead, communities shall agree on common definitions and formalize them through RDF. By publishing the definitions, for instance in the form of an ontology, others can adapt them and widen their acceptance.

RDF has a highly general information model. The basic description model is the triple: a subject (resource) linked to an object (another resource, or a literal value) through a property. Subjects and objects as nodes together with the properties as arcs make up a directed "description" graph. The result is a simple and uniform model: one and the same URI can be an arc and a node in the graph ([Champin, 2001]; see Figure 2.1 below).

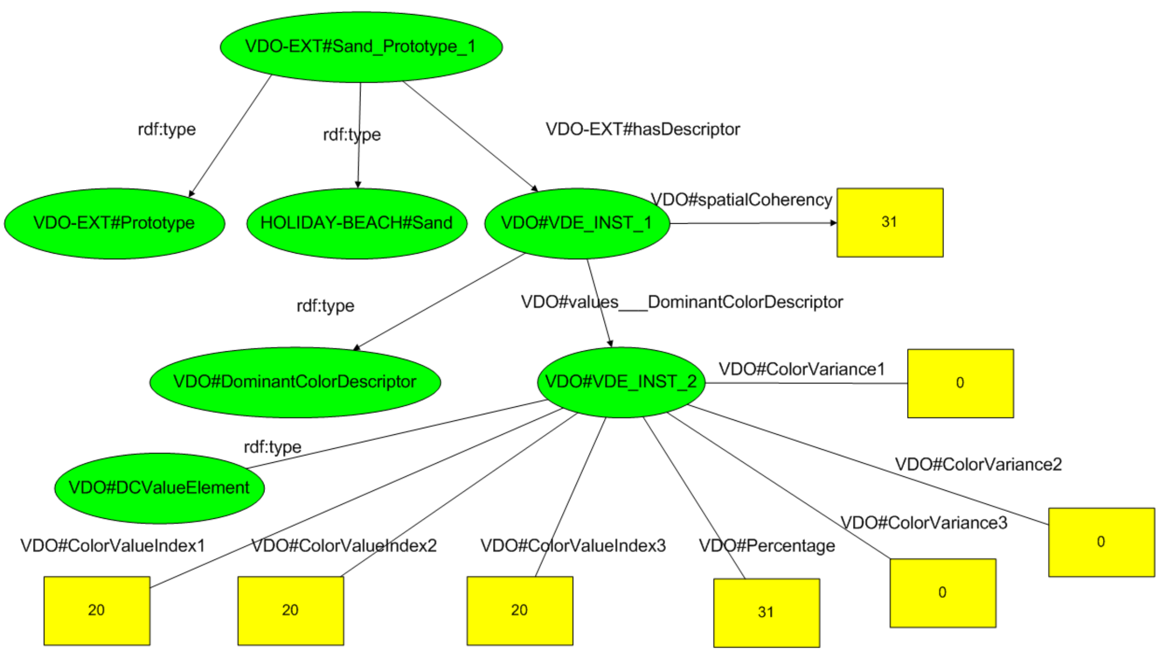


Figure 2.1. An example of RDF graph [W3]

RDF distinguishes three kinds of concepts:

i. Resources, properties and statements are fundamental concepts (rdf:Resource, rdf:Property, rdf:Statement). Statements are RDF triples in the form subject–predicate–object and resources, too.

ii. Schema definition concepts are used to define new RDF vocabulary. Available mechanisms include specialisation, categorization through class and type constructs, and limitation to domain and range. The former reduces the number of resources to which a property can apply. The latter controls the number of values a property can take on.

iii. Utility concepts are concepts that come in handy but are not essential for RDF. For example, collection properties and properties for comments belong to this category.

The following is a simple example for an RDF description in XML serialisation syntax.

```
<rdf:RDF
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
     syntax-ns#"
     xmlns:lib="http://www.zvon.org/library">

<rdf:Description about="RD">
     <rdf:type
     resource="http://www.zvon.org/library/Author"/>
     <lib:firstName>Roald</lib:firstName>
     <lib:surname>Dahl</lib:surname>
</rdf:Description>
...
<rdf:Description about="Matilda">
     <rdf:type
     resource="http://www.zvon.org/library/Book"/>
     <lib:creator rdf:resource='RD'/>
     <lib:pages>240</lib:pages>
</rdf:Description>

<rdf:Description about="The BFG">
     <rdf:type
     resource="http://www.zvon.org/library/Book"/>
```

```
        <lib:creator rdf:resource='RD'/>
        <lib:pages>208</lib:pages>
    </rdf:Description>
    ...
    </rdf:RDF>
```

The example is taken from [Nic, 2010] and shows books and their authors. The top-level element opens the description, at the same time declaring the rdf namespace for RDF language tags. The second referenced namespace (lib) declares some referenced description structures. The rdf:Description elements each describe a particular resource, whose URI (in this example: initials of authors and titles of books) is specified in the about attribute. rdf:Type expresses that the described resource is of the class that is defined at the URI given in the resource attribute. The rest of this description is straightforward. Please note that rdf:resource of the lib:creator tag points to a description that was just defined.

A more compact, “abbreviated” syntax is also available. It takes less space and can be embedded in HTML documents more easily. However, it lacks expressiveness [Ianella, 1998].

RDF was developed to enable the vision of the semantic web [Berners-Lee, et al. 2001] and plays a major role for its implementation. It has been enhanced with mechanisms to establish RDF vocabularies (RDF Schema, see [Brickley, et al 2004], can be used to establish ontologies and enables logical inferencing (DAML+OIL, see [Conolly et al., 2001]. RDF is easy to use. In the WWW, it uses a huge and established platform, and it is supported by the W3C. Thus, it reaches a big audience. It fulfils the requirements from above.

Still, there are several drawbacks. Similar to XML, where anyone can define new languages, RDF allows the definition of new properties. Unless one sticks to an existing RDF ontology, there is no gain in understandability in comparison to the use of XML. According to [Page et al. 2001], “RDF is not suitable for inter-operation of multimedia metadata since it has no linking mechanisms to spatio-temporal sections of data and limited data typing.” The recent standard MPEG-7 has been developed to address the special needs multimedia objects.

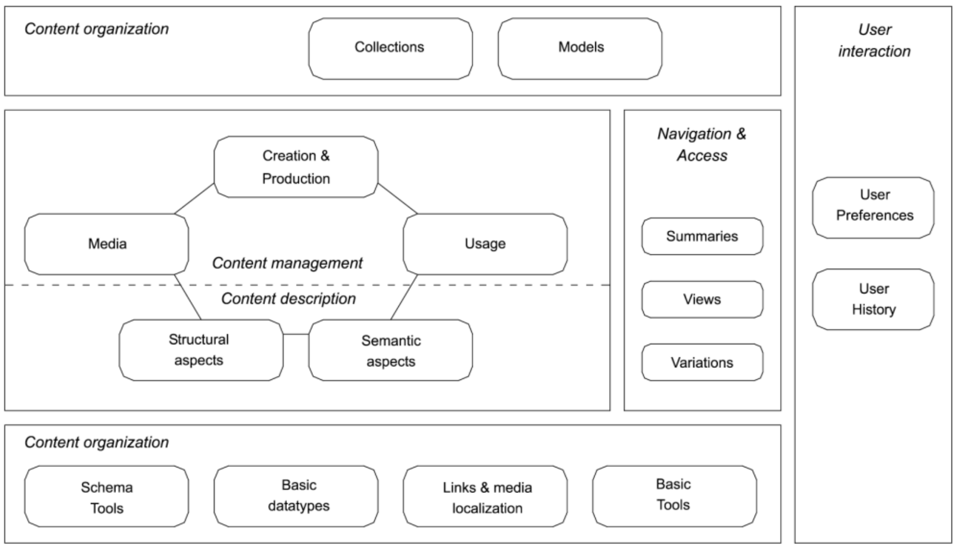
2.3.4 MPEG-7

[MPEG-7a] (Also known as multimedia content description interface) is a standard currently developed by MPEG (Moving Pictures Expert Group, see MPEG homepage). The amount of multimedia objects accessible to human end-users and automatic systems (e.g. agent technology) is steadily growing. MPEG-7 has been conceived to address the problem of finding relevant content in this mass. It applies to multiple forms of multimedia– among others, still pictures, graphics, 3D models, audio, speech or video, while covering the description needs of multiple domains.

Unlike the other standards (MPEG-1, MPEG-2, MPEG-4 and MPEG-21) of the group, MPEG-7 is no standard for content delivery, consumption and compression. MPEG-7 is defined as *“a standard for describing features of multimedia content.”* [Day, et al. 2002]. The goal is to increase interoperability between different vendors and reuse of metadata multimedia descriptions. The scope of MPEG-7 is only description of content. How the description is generated or accessed for search is not regulated. MPEG-7 descriptions describe content and form of multimedia material. They handle access rights and provide classifications of the described material. They can specify context and can link to unseen relevant content. MPEG-7 combines and builds on existing standards, and is designed to interoperate with them. It shall enable search for multimedia content by humming melodies, drawing sketches or outlining movie plots. [Day, et al. 2002] claim *“MPEG-7 provides the world’s richest set of audio-visual descriptions.”*

The Figure 2.2 shows the pictorial representation of the MPEG-7 multimedia description schema. The MPEG-7 descriptions are XML documents corresponding to the MPEG-7 schemas. The standard consists of the following elements:

i. *Descriptors* (D) are the basic unit of an MPEG-7 description. They are meant to describe low-level features – e.g. location, time or quality - and are expected to be extracted from the material automatically. A Descriptor's role is similar to an element or tag in an XML file.



Source: Martinez (1999)

Figure 2.2: Overview of the MPEG-7 multimedia description schemas

ii. *Description Schemas (DS)* are composite objects. They consist of and organize the relationships between their components: Descriptors or Description Schemas. Description Schemas aim at higher-level features of content and are usually annotated by humans. DSs describe for instance regions, objects and events. A Description Schema resembles the functionality of a XML DTD. Description Schemas and Descriptors are also subsumed under the term description tools.

iii. *The Description Definition Language (DDL*) is used to define new and modify or extend old Descriptors and Description Schemas. It is made of XML Schema with extensions required for the description of audio-visual content. Namely, array, matrix and primitive time data types were added [Martinez, 2002].

iv. *System tools:* MPEG-7 descriptions are usually in textual, tagged form. The associated overhead is inefficient for transmission and storage of descriptions. The

MPEG group is developing an alternative, binary format for descriptions (BiM), "transmission mechanisms (both for textual and binary formats), multiplexing of descriptions, synchronization of descriptions with content, management and protection of intellectual property in MPEG-7 descriptions, etc." [Martinez, 2002]

The claim of exhaustiveness for multiple domains entails a huge engineering effort. Therefore, development has been split into several parts. Each of them forms one fraction of the standard. [MPEG-7b] The different parts are

i. *MPEG-7 Systems* – the tools needed to prepare MPEG-7 descriptions for efficient transport and storage and the terminal architecture.
ii. *MPEG-7 Description Definition Language* - the language for defining the syntax of the MPEG-7 Description Tools and for defining new Description Schemes.
iii. *MPEG-7 Visual* – the Description Tools dealing with (only) Visual descriptions.
iv. *MPEG-7 Audio* – the Description Tools dealing with (only) Audio descriptions.
v. *MPEG-7 Multimedia Description Schemes* - the Description Tools dealing with generic features and multimedia descriptions.
vi. *MPEG-7 Reference Software* - a software implementation of relevant parts of the MPEG-7 Standard with normative status.
vii. *MPEG-7 Conformance Testing* - guidelines and procedures for testing conformance of MPEG-7 implementations
viii. *MPEG-7 Extraction and use of descriptions* – informative material (in the form of a Technical Report) about the extraction and use of some of the Description Tools.
ix. *MPEG-7 Profiles and levels* - provides guidelines and standard profiles.
x. *MPEG-7 Schema Definition* - specifies the schema using the Description Definition Language

2.4 Methods for Multimedia Annotation

Applications such as social media, distance learning, digital libraries, video-on-demand, online images storage, digital video broadcast, interactive TV, multimedia information systems generate and use large collections of digital data. This has caused a need for tools that can professionally catalogue, search, browse and retrieve related material. Enormously research has been conducted for images annotation as the images is the simplest form of multimedia and include only one modality i.e. *visual modality*, while video is the most complex form of multimedia as it's a combination of multimodal (*Textual*, *Visual and Auditory*). Despite the possibility that multimodal processing methods have been shown to be efficient in specific applications, we cover only visual modality of the video as most of the annotation processes for visual modality of the videos are taken from the images. Temporal video segmentation is the first step towards annotation of videos. Its purpose is to break up the video into a set of meaningful and manageable segments (shots). Each shot is then represented by selecting key frames. Each key frame is the visual representation of the shot and is treated similar to image for annotation. There are numerous annotation techniques for multimedia, we have categorized them into manual, semi-automatic and automatic annotation.

2.4.1 Manual Annotation

This is the "old-fashioned" approach where people have non-digital paper pictures in photo albums and write the associated text. Manual annotation is a completely human oriented task that deals with human oriented information. This type of metadata can be the event of the image, the photographer, the title and similar information. The advantage of manual annotation is the accuracy in extracting semantic information at several levels. It is the most precise way of annotation and for now, the only way of full value to add semantics to images.

Manual annotation is manageable for small multimedia collections, but for larger digital collections it is far too time consuming to annotate each single multimedia file in the collection and this is the biggest disadvantage of manual annotation [Jack. et al 2005; Kerry, et al. 2003]. The investigation done by [Kerry, et al. 2003] shows some of the users' behavior regarding their personal digital image collections. Images / videos are downloaded from the camera, labeled with a software-generated name and placed in a folder. The name automatically generated by the camera software most often consists of letters and digits that do not have any semantic value. Most users do not interfere with the software's decisions then and to change the name of the images / video later on, is a task that is most often not carried out.

Another snag is that the task of illustrating the content of digital contents is highly subjective. The standpoint of textual descriptions given by an annotator could be distinctive from the perspective of a user. An image can mean different things to different people and is more complicated in case of video. It can also mean unusual things to the same person at different times.

Even with the same perspective, the words used to describe the content could vary from one person to another. In other words, there could be a variety of discrepancies between user textual queries and multimedia annotations or descriptions [Chen, et al. 2005]. To be able to compose a query that will result in relevant images, the annotator and retriever must have some common vocabulary and a common understanding of the world. If the annotated text and the query-text are completely different this might return no relevant results even if they potentially exist. Based on the work of [Jack. et al 2005], we believe it is naive to think that users will manually annotate large image collections if they are given other options [Jack. et al 2005; Keller et al. 2004] –and even if they are not!

The Video Image Annotation Tool [VIA], VideoANT [ANT] and [LabelMe] are tools to manually annotate videos and images. They provide a user friendly interface for the accurate and undemanding live and *"frame by frame"* annotation of video and still images. Similar approach has been adopted by YouTube, but its services only available for videos, while [Flicker] provide the same features for videos and images, but there is no support for *"frame by frame"* annotation. The [SpiritTagger], [Alipr] and Advanced Image Annotation (AIA) Tool are the best tools for manually image annotation. Moreover, [Anvil] is a publicly available research tool for exclusively manual video annotation, where the annotation scheme is generic and customizable. Customization can be done by specifying a set of attribute-value pairs which are used to attach the metadata. The structuring possibilities are simple definitions of annotateable frame sequences. Its original purpose was to annotate gesture and speech semantics in videos. The M-OntoMat-Annotizer [MOA] is a public semantic annotation tool that was developed in the context of the aceMedia project [aceMedia]. Basically, it enables the user to attach metadata to videos or images. The basic idea of the tool is to extract low-level MPEG-7 descriptors and link them automatically to ontologies and semantic annotations in order to annotate high level semantics. The VideoAnnEx annotation tool [VAE] allows the user to annotate shots in a video. The annotation data is stored in an MPEG-7 file. Each shot in the video can be annotated with static scene descriptions, key object descriptions, event descriptions, or other lexicon sets of descriptions. This restricts the annotation possibilities to the content of the lexica but keeps the annotations simple and consistent.

2.4.2 Automatic Annotation

Automatic annotation is machine annotation, where humans only verify the task. The information added by a camera is of a technical nature and is automatically added. This information is typically time, location, resolution quality, camera model, which number the file has in the range of images /videos taken, name of the image/video and other technical information. As we see from this type of information automatic annotation is limited due to computers lacking ability to extract semantic information from these kinds of multimedia objects. Even in an ideal world where face recognition and shape detection works perfectly, a computer will not be able to abstract event information like *"The 5th birthday party of Lutfullah"* or other deep semantic information [Jack. et al 2005]. There are several situations where the images and/or videos are automatically generated and have minimum of information attached. A surveillance camera may take series of photos or even record a video and store them in a database without any human interaction. The footage might be stored in folders annotated with the actual date. Specific images / videos of a specific event will then be impossible to retrieve without browsing the footage collection. To annotate each object in such a collection would be useless. We divide the automatic annotation approaches into supervisor/un-supervisor techniques.

2.2.2.1 Un-Supervisor Techniques

Unsupervised learning methods for image annotation have a common characteristic, i.e., they view keywords as a type of feature, i.e. textual features, so that they are distinguished from the visual features. We divide these approaches further into two categories, i.e., parametric approach and non-parametric approach. All of the parametric approaches have a training stage to estimate the parameters. In contrast, non-parametric approaches do not need to estimate any parameters in the training stage, but they do need the whole training data whenever they are used to annotate a new image.

a) Parametric Approach

The first attempt at automatic image annotation by viewing words as textual features is perhaps the work of [Mori, et al. 1999], in which they proposed a co-occurrence model to represent the relationship between keywords and visual features. Each image is converted into a bag of rectangular image regions obtained by a regular grid. The image regions from the training data are clustered into a number of region clusters. For each training image, they propagate its keywords to each image region in this image. The conditional distribution of keywords of each region cluster can be estimated from the empirical distribution on the training data. Given a new image, the conditional keyword distribution of each individual image region is aggregated to generate the conditional keywords distribution of the test image. Figure 2.3 and Figure 2.4 illustrate the training and test process of the co-occurrence model proposed in [Mori, et al. 1999] respectively.

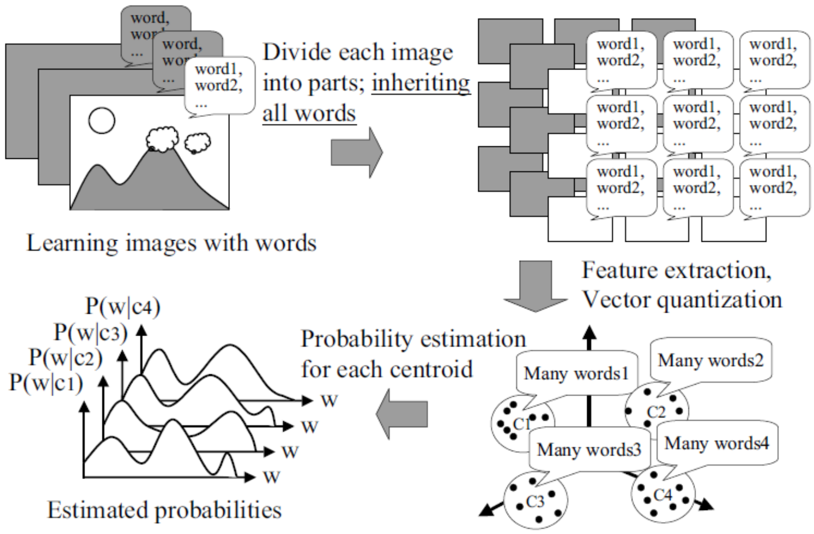


Figure 2.3: The training process of the co-occurrence model [Mori, et al 1999]. The keywords annotated to a training image propagated to each rectangular region in the image with equal chances.

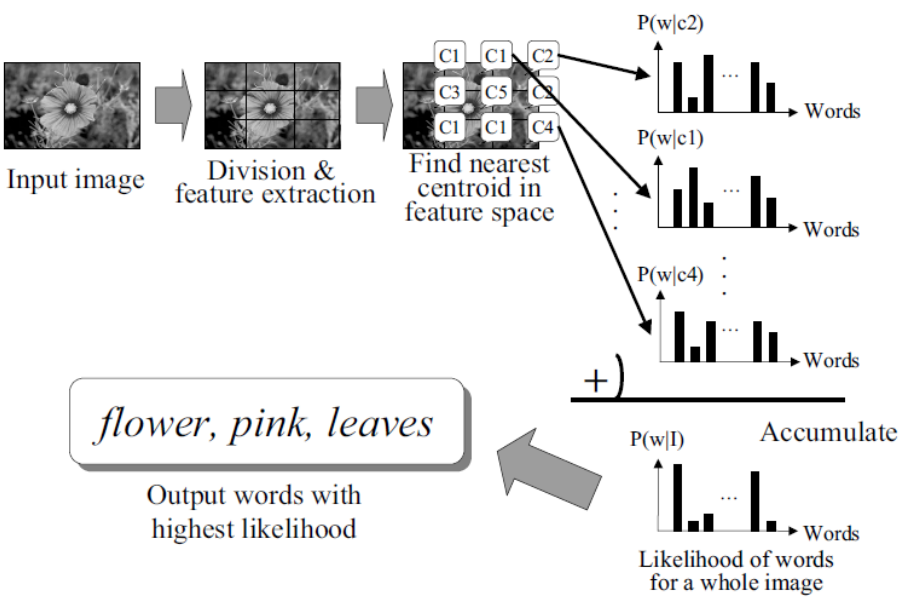


Figure 2.4: The test process of the co-occurrence model [Mori, et al 1999]. The keyword distributions of all the rectangular regions are aggregated to generate the keyword distribution of the whole image

The major drawback of the above co-occurrence model is that it assumes that if some keywords are annotated to an image, they are propagated to each region in this image with equal probabilities. This assumption is violated in many real situations because many keywords are object names such *as "sky", "sun"* and *"water".* The appearance of this kind of concept in an image is usually a small portion of an image instead of the whole image. Thus, [Duygulu et al. 2002] proposed a machine translation model for image annotation, which is essentially an improvement of the co-occurrence model of [Mori et al. 1999]. They represent an image as a bag of image regions obtained by image segmentation and performed vector quantization on each of these region features. The vector quantized image regions are treated as "visual words" and the relationship between these and the textual keywords can be thought as that between one language, such as French, to another language, such as German. The training set is analogous to a set of aligned bitexts, i.e. texts in two languages. Given a test image, the annotation process is similar to translating the visual words to textural keywords using a lexicon learned from the aligned bitexts. They found that a relatively simpler translation model used in the language translation, i.e. the model of [Brown et. al 1990] produced better performances than other available language translation models. Similar to the co-occurrence model [Mori, et al 1999], the learned parameters of the translation model are also the conditional distribution probability table, but the translation model does not propagate the keywords of an image to each region with equal probability. Instead, the association probability of a textual keyword to a visual word is taken as a hidden variable and estimated by an Expectation-Maximization (EM) algorithm [Dempster, et. al 1977].

A similar approach to the above machine translation model is to use a hidden Markov model (HMM) [Lawrence, et Al 1989] proposed by [Ghoshal et al. 2005]. In this approach, each textual keyword is represented by a hidden state, which can generate visual features following as per state probability distribution. The training process aims to find the best correspondence of image regions and textual keywords and estimate the parameters for each state. The annotation process of a new image is equivalent to recovering the most likely hidden state of each image region. A major difference between the HMM approach and the machine translation model is that the HMM approach models the continuous distribution of visual features, whereas the translation model represents the keyword distribution of each vector quantized image region. However, the HMM model assumes a transition process between different states (textual keywords) which is not necessarily supported by real data.

Instead of modelling the conditional distribution of textual keywords based on visual features, some researchers proposed methods to model the joint distribution of textural features and visual features. One such attempt is made by [Barnard, et al. 2001]. They define a document as a combination of visual features and textual features. A hierarchical factor model is proposed to model the joint distribution of textual features and visual features, as illustrated in Figure 2.5. The model assumes that a document belongs to a cluster, which is denoted by the leaf nodes in the tree hierarchy. Given the document and the cluster it belongs to, the document is generated by the aspect nodes on the path from the root node to the leaf node following the hierarchical structure (see the arrows in Figure 2.5). Each aspect on the path can generate image regions and textual features following a per aspect probability distribution. Since different clusters have distinct traversing path, each has a separate joint models of the aspects for each other. Moreover, since all the aspects are organized in a hierarchical structure, the aspects are very compact and it can model the commonalities between clusters in different degrees between. However, this model is optimized for image clustering instead of linking textual words to image regions. [

Zhang et al. 2005] proposed a probabilistic semantic model to represent the joint distribution of the image features and the textual words. They assume that there are a number of hidden semantics in an image, each semantic has a probability to generate the global visual feature and the textual words. Given a specific semantic, the generation of visual features and textual words are independent from each other. The major difference between this approach and [Blei, et al 2003] and [Barnard, et al. 2001] is that it takes an image as a whole instead of a set of regions.

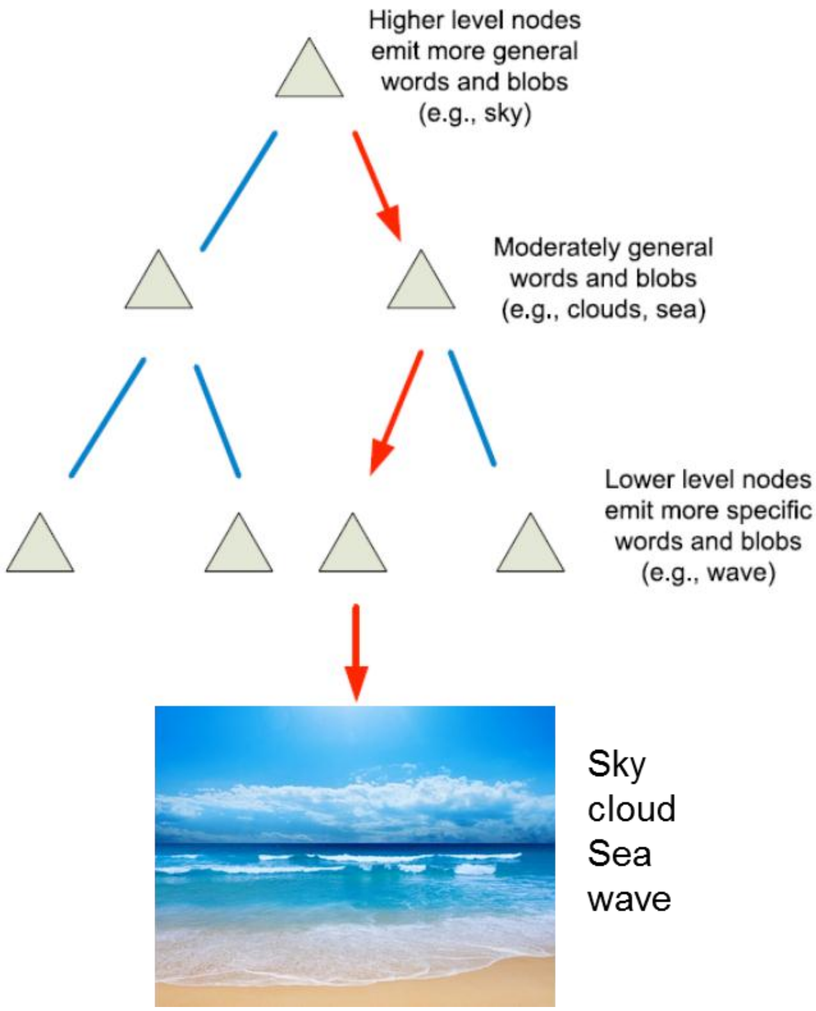


Figure 2.5: The hierarchical aspect model of Barnard and Forsyth [Barnard, et al. 2001]. Each triangular node represents an aspect. The higher level nodes generate general visual features and textual features whereas the lower level nodes generate specific visual features and textual features. An image belonging to a specific document cluster is generated by all the nodes on the transversing path (see the red arrows in the figure) from the root node to the leaf node.

[Monay, et al. 2004] explored latent semantic analysis (LSA) [Scott, et al. 1990] and probabilistic latent semantic analysis (PLSA) [Hofmann, 2001] for automatic image annotation. In short, a document of image and texts can be represented as a bag of words, which includes the visual words (vector quantized image regions) and textual words.

Then LSA and PLSA can be deployed to project a document into a latent semantic space. Annotating images is achieved by keywords propagation in this latent semantic space. The original LSA model is linear, so Liu and Tang [Liu, et al. 2005] proposed an extension of the LSA method to non-linear LSA. Since LSA and PLSA essentially model the co-occurrence relationship between any words including the textual words and visual words, it does not focus on the co-occurrence relationship between textual words and visual words. In most cases, the number of textual words (1 " 5) is very small compared to the number (200 " 300) of visual words in an image. So many efforts have been made on modelling the co-occurrence between visual words, resulting in relative low discriminative capabilities.

The above mentioned parametric models are equivalent to taking an abstract from the training data, i.e. the complexity of the model itself is only dependent on the number of parameters to be estimated. However, the estimation of model parameters usually relies on an E-M algorithm, in which only a local optimum of the estimated parameters can be achieved and its capability of discriminating different concepts is limited.

b) Non-parametric Approach

Different from a parametric model, a non-parametric model does not have a training process. [Joen et al. 2003] formulated the problem of automatic image annotation as cross-lingual information retrieval and have applied the cross-media relevance model (CMRM) to image annotation. Although CMRM also tries to model the joint distribution of visual features and textual words, it is a non-parametric model, like the *k*-NN [Duda, et al. 2001] approach for pattern classification. The essential idea is that of finding the training images which are similar to the test image and propagate their annotations to the test image. CMRM does not assume any form of joint probability distribution on the visual features and textual features so that it does not have a training stage to estimate model parameters. For this reason, CMRM is much more efficient in implementation than the above mentioned parametric models. A drawback of the CMRM model is that it vector quantized the image regions into image blobs and this can reduce discriminative capability of the whole model. So [Manmatha et al. 2004] have proposed an improved model, i.e. the continuous cross-media relevance model (CRM). CRM preserves the continuous feature vector of each region and this offers more discriminative power.

[Feng et al. 2004] proposed a further extension of the CRM model called the multiple Bernoulli relevance model (MBRM). They suggest that the assumption of a multinomial distribution of keywords in CRM [Manmatha, et al. 2004] and CMRM [Jeon, et al 2003] favors prominent concepts in the images and equal length of annotation for each image. So they proposed to model the keyword distribution of an image annotation as a multiple Bernoulli distribution, which only represents the existence/nonexistence binary status of each word. Their experimental results show that MBMR outperforms CMRM [Jeon, et al 2003] and CRM [Manmatha, et al. 2004] for the annotation of video frames, in which the annotation length of each image varies a lot and the most important issue is the existence of a concept rather than its prominence.

All the above mentioned methods predict each word independently given a test image. They can model the correlation between keywords and visual features but they are not able to model the correlation between two textual words. To solve this problem, [Jin et al. 2004] proposed a coherent language model which is extended from CMRM [Feng et al 2004]. The model defines a language model as a multinomial distribution of words. Instead of estimating the conditional distribution of a single word, they estimate the conditional distribution of the language model. The correlation between words can be explained by a constraint on the multinomial distribution that the summation of the individual words distribution is equal to one. Thus the prediction of one word has an effect on the prediction of another word.

[Pan et al. 2004] proposed a graph-based approach (GCap) for automatic image annotation. They represent an image as a set of regions, each of which is described by a visual feature vector. A graph is constructed on the whole training data. They define three types of node in this graph: 1) image node representing an image, 2) region node representing an image region, 3) word node representing a textual keyword. The links between nodes represent the relationship between different units (image, region and words), these include: 1) image attribute link, which connects an image to its keywords and its visual features nodes, 2) region links, which connect each region node to its *k* nearest neighboring region nodes. The idea of GCap can be illustrated by the graphical model in Figure 2.6. Given a test image, the image regions are obtained by unsupervised image segmentation. An image node representing the test image and several region nodes representing the image regions in the test image are added to the graph constructed on the training set. Since the textual words of the test image are missing, there is no direct links between the test image nodes to any of the keyword nodes. The annotation process is modelled as a random walk with restarts (RWR) [Tong, et al. 2006] on the graph. The steady state probability of a random walk to arrive at a textual word node from the test image node is the annotation probability of this word to the image. Similar to the CMRM model, this approach is also a non-parametric model. Since it needs to store the training data in a graph structure, it is not efficient and is not applicable to applications that involve a large dataset.

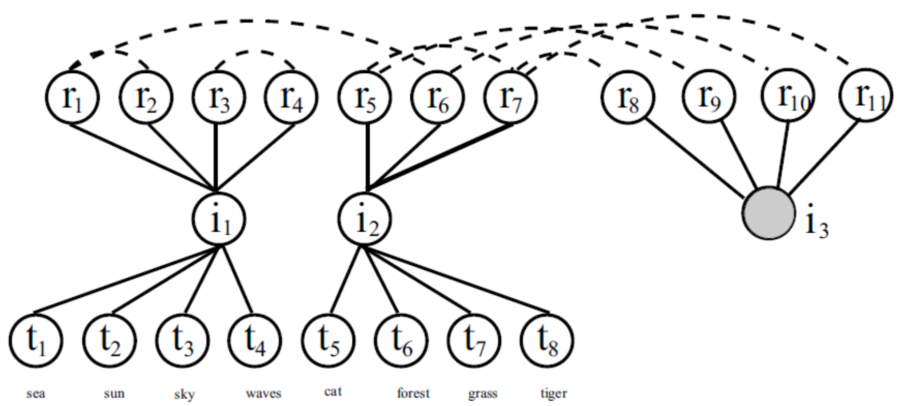


Figure 2.6: The GCapmodel of [Pan et al. 2004]. The image nodes ($\boldsymbol{i_1}, \boldsymbol{i_2}$) are connected to its region nodes ($\boldsymbol{r_i}$) and textual word nodes ($\boldsymbol{t_i}$). To annotate an un-annotated image ($\boldsymbol{i_3}$), a random walk starts from $\boldsymbol{i_3}$. The steady probability of the random walk to reach a textual word ($\boldsymbol{t_i}$) is taken as the probability of annotating $\boldsymbol{t_i}$ to $\boldsymbol{i_3}$.

[Liu et al. 2006] proposed an adaptive graph model for image annotation. They also construct a graph on the training data. But unlike GCap [Pan et al. 2004], there is only one type of node, i.e. the image node. Each image node is connected to its k nearest neighbors. The graphical model is adaptive in the sense that the number of nearest neighbors connected to each image node, k, is different to each other and decided by an adaptive process. The similarity between two image nodes is a weighted global visual similarity. The annotation probability of each word to the images is represented in a ranking order matrix. For a un-annotated image, the ranking order matrix is obtained by iteratively updating the matrix by the manifold ranking algorithm [Zhou, et al. 2004].

In summary, unsupervised learning based methods have their advantages: they make an assumption of a model which can express explicitly the complex relationships between textual words and visual features by incorporating available a prior information. What is more, some approaches, such as the co-occurrence model [Mori, et al 1999] and the translation model [Duygulu, et al 2002], can even associate a word to each region in an image. This annotation-by-region strategy is more informative than annotating an image as a whole. However, most of the unsupervised learning based methods rely on an E-M procedure for training. The E-M procedure is sensitive to the initial parameters and with its complex objective function it can only produce a local optimum solution, which in turn leads to inferior performance of the model to unseen data. For the non-parametric models, such as CMRM [Jeon, et al 2003], they need to store the whole training data in the annotation system, which is not desirable for large database. Also, non-parametric models assume that a perfect set of data are available to be used as the reference set, which is not usually the case.

2.2.2.2 Supervisor Techniques

Besides considering the keywords annotated to images as a kind of features as that in the unsupervised methods, we can also view them as different class labels. By doing this, the process for annotating an image with a keyword becomes similar to that of classifying the image as to whether it belongs to a particular class. This is the underlying motivation of image annotation based on image classification.

It is worth noting that although image annotation emerged as an active topic only in the last decade, the problem of image classification has a much longer history. In the early days before 1990's, image classification mainly focused on some special image domains, such as synthetic aperture radar (SAR) images [Ulaby, et al. 1986], medical images [Chen, et al. 1986], multi-spectral images [Kettig, et al. 1976], remote sensing images [Kirvida, et al. 1976], industrial inspection [Capson, et al. 1988] etc. It is only in recent years that attention has begun to be paid to general images such as consumer photographs, perhaps because such types of images are made more easily available with the rapid progress in the quality of imaging device. More recently, automatic image annotation has been linked to image classification and in most cases, its goal is to provide viable indexing and retrieval of the images in large image databases.

Existing approaches to image annotation based on image classification fall into three categories:

a) Global scene-oriented classification methods which extract a global feature descriptor from an image and then deploy a statistical classifier for image classification. Examples of this kind of class label include *"countryside", "landscape", "outdoor"* and so on. The task is usually classifying the image as a whole. Figure 2.7 illustrates such kind of image annotation system.
b) Local object-oriented classification methods which classify images by object names. The image content assigned to the labels is usually a part of the image. Examples of these class labels include *"balloon", "water", "people"* and so on.
c) Multi-level classification methods which assign class labels in a hierarchical structure, including both scene-oriented class and object-oriented class.

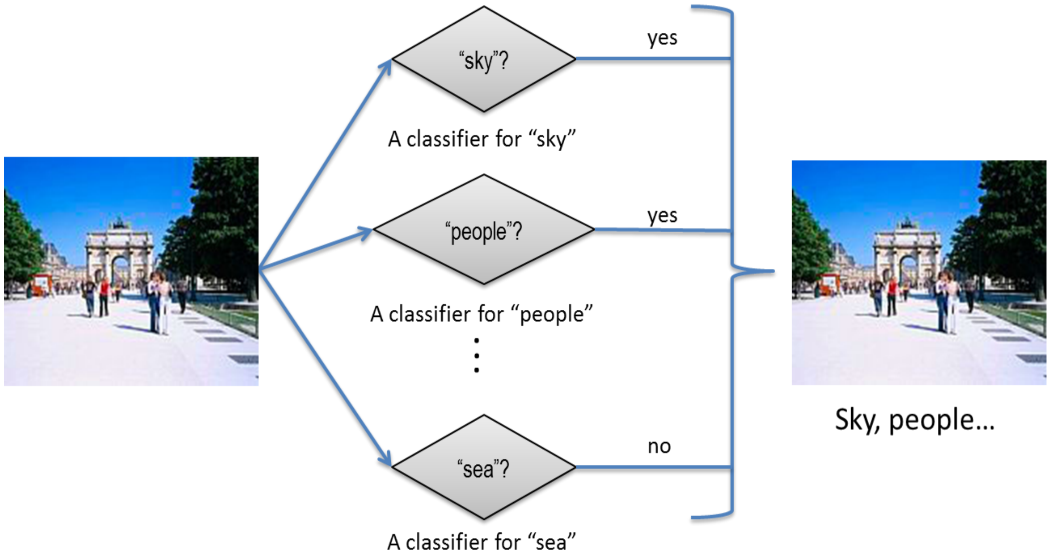


Figure 2.7: An illustration of the image annotation system through image classification. Each concept can have an independent image classifier.

a) Global Scene Oriented Classification

Among the global scene-oriented classification approaches, some of the early work focused on designing visual features. For example, [Gorkani et al. 1994] were able to classify *"city"/"suburb"* images by using global multiscale orientation features. [Lipson et al. 1997] made an attempt to incorporate quantitative spatial and photometric relationships within and across regions in low resolution images (such as 20×20 pixels) for natural scene image classification. They hand-crafted the template used to describe the spatial and photometric relationships for each scene class. Later, [Ratan, et al. 1997] proposed a similar classification method as that in [Lipson, et al. 1997] but learned the configuration templates of each class from a few human selected training examples. [Huang et al. 1998] have proposed a hierarchical image classification scheme. They used color correlograms [Huang, et al. 1997] as the visual features and a classification tree as the classifier. In a related work, [Vailaya et al. 1998] examined the discriminative capability of different visual features for *"city"* vs. *"landscape"* scene classification and have found that the edge direction-based features have the best discriminative capability on their dataset. By focusing on various visual features the approaches provide a good basis for the following work of image classification, but the statistical classifiers are not powerful enough.

With rapid progress in the machine learning community, more and more powerful statistical classifiers have become available, such as the support vector machines (SVM) [Nello, et al. 2000]. Thus, recent work pays more attention to exploiting statistical classifiers and more powerful visual features at the same time. [Chapelle et al. 1999] have attempted to solve the general image classification problem using SVM's [Nello, et al. 2000] and have used an enhanced heavy-tailed RBF kernel for high dimensional image features. [Fung, et al. 1999] decompose the semantics of a scene image into two levels: (1) the primitive semantics at the patch level, and (2) the scene semantics at the image level. The learning of primitive semantics is based on a supervised clustering of the patch features. Their scene classification is achieved by using the distribution of each primitive in an image.

Scene semantics are made more explicitly by [Vailaya et al. 2001] who proposed a method for hierarchical classification of vocational images: at the highest level, images are classified as *"indoor"* or *"outdoor"; "outdoor"* images are further classified as *"city"* or *"landscape";* finally, a subset of *"landscape"* images is classified into *"sunset", "forest",* and *"mountain"* classes. They model the probability density of each scene class through vector quantization [Gray, et al. 1986] and classify images based on the maximum a posterior criterion. [Chang et al. 2003] proposed a soft categorization method of images based on the Bayes point machines (BPM) [Herbrich, et al. 2001], which is another advanced kernel based classifier.

The above mentioned methods are based on global visual features extracted from a whole image. [Wang, et al. 2002] proposed an image categorization method based on using the 2D multi-resolution hidden Markov model (2D-HMM) [Lawrence, 1989]. Images are segmented into regions by employing a multi-resolution regular grid. 2D-HMM can model the dependency between regions in the same resolution and the regions across different resolutions. [Csurka et al. 2004] proposed a bag-of-keypoints model for object class categorization. Each image is represented as a bag of salient regions obtained by interest point detectors. Each region is represented by a visual feature vector. After a vector quantization process on the region features, an image can be taken as a bag of visual words. The frequency vector of visual words is taken as the global feature vector and a SVM classifier is deployed to classify images of object classes. [Carneiro, et al. 2005] proposed an image annotation framework based on hierarchical mixture modelling of the probability density estimation of each class. Each image is represented as a set of patch features. The distributions of these patch features for each concept is modelled as a Gaussian mixture model and all the concepts are modelled by a hierarchical Gaussian mixture model (Hier-GMM). Their experimental results show that the Hier-GMM is efficient for large database. [Maree et al. 2005] proposed an image classification method by combining the random sampling of sub-window images and an ensemble of extremely randomized tress. Since they have added various transformations in the process of abstracting random sub windows, their approach is robust to both scale and rotation, however they have not tested their approaches on a more complex image dataset for image annotation.

Combining complementary features can produce successful results. [Datta et al. 205] proposed a generic image categorization system based on two heterogeneous generative models one per image category. The two models provide evidence for categorization from two different aspects of images, i.e. a structure-composition (S-C) model constructed from the Beta distribution to capture the spatial relationship among segmented regions of images, and a Gaussian mixture model of color-texture (C-T) features. The top N independently predicted annotation evidences by these two models are further refined by taking into account the word frequency, word salience and word congruity based on WordNet [Miller, 1992]. This combination of a structure and non-structure model offer more discriminative power for generic image categorization compared to other approaches using only one of these two types of models.

Some methods are based on sophisticated probabilistic models. [Li, et al. 2006] represent each image as a probabilistic distribution of color and texture features. Each image category is modeled as probabilistic distribution of probabilistic distributions. Taking advantage of the fast optimization algorithm, their approach can achieve real time annotation performance on a large scale dataset. However, it is not clear how well their method can perform on individual object concepts.

Among the work of image classification, some recent work focuses on classifying a very small set of concepts, such as natural scene categories. They can be further divided into two categories. The first relies on self-defining the intermediate features. [Oliva, et al. 2001] proposed a set of perceptual dimensions (naturalness, openness, roughness, expansion and ruggedness) that represent the dominant spatial structure of a scene. Each of these dimensions can be automatically extracted and scene images can then be classified in this low-dimensional representation. [Vogel, et al. 2007] used the occurring frequency of different concepts (water, rock, etc) in an image as the intermediate features for scene image classification, and they need manual labelling of each image patch in the training data. Whereas manually labelling can improve the semantic interpretation of images, it is still a luxury for a large dataset and it can also introduce inconsistencies in how the common set of concepts are defined. [Vogel, et al. 2007], the second kind of approach is aimed at alleviating this burden of manual labelling and learns the intermediate features automatically. This is achieved by making an analogy between a document and an image and taking advantage of the existing document analysis approaches. For example, [Fei-Fei, et al. 2005] proposed a Bayesian hierarchical model extended from latent Dirichlet allocation (LDA) to learn natural scene categories. [Bosch et al. 2006] achieved good performance in scene classification by combining probabilistic latent semantic analysis (PLSA) [Hofmann. 1999] and a KNN classifier. A common point of these approaches is that they represent an image as a bag of orderless visual words. An exception is the work done by [Lazebnik, et al. 2006] where they proposed spatial pyramid matching for scene image classification by partitioning an image into increasingly fine sub-regions and taking each sub-region as a bag of visual words.

These examples of global scene-oriented image classification have been proved to be effective in classifying many scene categories, such as *"sunset", "landscape"* and *"countryside",* but they have not shown any advantage in classifying object names, such as *"sky", "tiger", "horse"* etc.

b) Local Object Oriented Classification

For individual objects, the corresponding visual appearance in the image is usually a segment of the image instead of the whole image. Sometimes, even collectively, these object segments may only make up a small part of an image. This makes a global visual feature not always an appropriate solution, especially in the case of heavy background clutter or when a number of different objects exist in the image. Therefore, treating an image as a bag of image regions and annotating image by these regions is helpful for the object-based classification of images.

Image annotation can be formulated as a multiple instance learning (MIL) problem as described by [Dietterich, et al. 1997]. In the MIL setting, the object to be classified is a bag of instances instead of a single instance. The training data is a set of positive bags and negative bags. A bag is labelled as positive if at least one of the instances in the bag is labelled as positive. A bag is labelled as negative if none of the instances in the bag is labelled as positive. This concept of positive bags and negative bags is illustrated in Figure 2.8. The labels on the training data are only provided for each bag, not for each instance. Given a new unlabelled bag, we need to classify it as positive or negative. This kind of problem cannot be solved by traditional statistical classifiers where each training example or test sample is represented as a single feature vector instead of a bag of feature vectors.

A number of approaches have been proposed based on the above MIL formulation of image annotation. [Maron, et al. 1998] made the first attempt at applying MIL techniques to natural scene image classification, distinguishing between terms such as *"sky", "waterfall"* and *"mountain".* They represent each image as a bag of sub images of 2×2 pixels, each of which is represented by a feature vector containing the mean color and the color difference between itself and its four neighboring sub images. The training of this MIL is through maximizing the diverse density (DD), i.e. search for the point in the instance feature space which is close to at least one of the instances in each of the positive bags and far from all the instances in each of the negative bags. Later, [Yang, et al. 2000] applied MIL to image annotation with the objective of explicitly annotating individual image regions instead of just labeling the whole image. They use the point-wise diverse density (PWDD) algorithm to find the corresponding image regions in the training set for a concept. Compared to the traditional DD algorithm of [Maron, et al. 1998], the optimal DD point that PWDD found is always an image region from the training set.

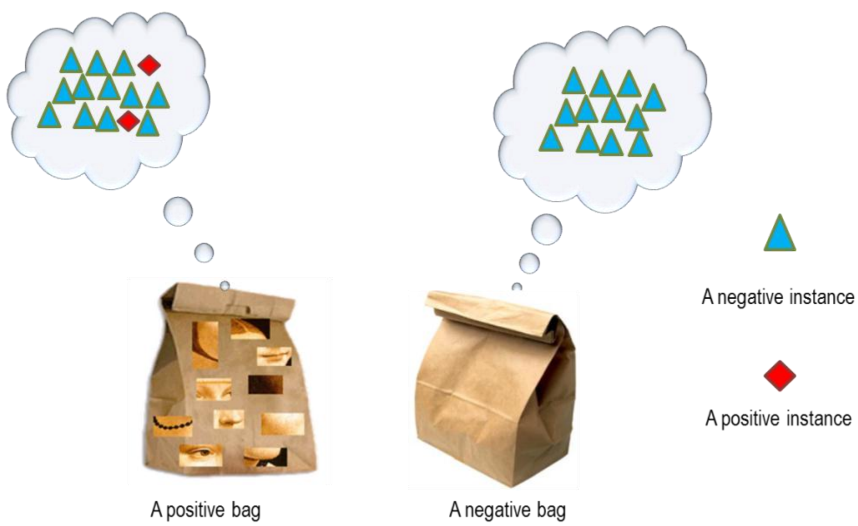


Figure 2.8: Bags and instances in multiple instances learning (MIL). A positive bag contains at least one positive instance. A negative bag contains no positive instance. The problem of MIL is classifying new bags given only the positive/negative labels of the training bags, without knowing the label of individual instances in each bag.

The above DD algorithm is computationally expensive, so other training algorithms for MIL have been proposed and applied to image annotation. [Andrews, et al. 2003] formulated the MIL problem as a mixed integer quadratic program. In their formulation, integer variables are the selector variables that indicate which instance in a positive bag is a positive instance. Their algorithm, which is called MI-SVM, has an outer loop and an inner loop. The outer loop sets the values of these selector variables. The inner loop then trains a standard SVM in which the selected positive instance replaces the positive bags. The MI-SVM approach is prone to becoming stuck into a local optimum solution which can affect the performance of the final classifier. So [Yang, et al 2006] proposed an asymmetric support vector machine method (ASVM) to solve the MIL problem and have applied it to region-based image annotation. Their method, which is called ASVM-MIL, extends the conventional support vector machines to the MIL setting by introducing asymmetrical loss functions for false positive and false negatives. Since this is an extension of the traditional SVM, the training algorithm can be formulated as a standard quadratic programming problem which is very efficient.

Apart from these attempts that use different training algorithms for the MIL algorithm, [Chen, et al. 2004] argue that some concepts cannot be described by a single instance in a bag, which is the basic assumption of the traditional MIL algorithm. Instead, these concepts can be only described by a combination of different instances. For example, a *"skiing"* scene means a combination of *"people"* and *"snow"*. For this reason, they proposed an algorithm, called diverse density support vector machine (DD-SVM), to learn the multiple aspects of a concept. DDSVM goes in two steps: in the first step, a set of prototypes are identified by the DD algorithm of [Maron, et al. 1998], each prototype is a local maximiser of the DD optimization function. In the second step, they map each bag of instances to a feature vector of fixed length using the distances between each instance in the bag to the set of prototypes. After obtaining this feature vector of fixed length, a traditional SVM classifier is applied to classify this new example of the vector feature space.

c) Multi-level Classification

Both the global scene-oriented classification and the local object-oriented classification approaches are advantageous for dealing with certain types of image categories. However, we are often faced with the problem of annotating images that contains both global scene-oriented class and local object-oriented class elements. So we need a comprehensive approach which can annotate these two types of class together.

Since the categorization of images by human tends to follow a hierarchical structure [Vailaya, et al. 2001], a multi-level classification scheme is likely to be helpful. For example, we may classify an image as a *"garden"* image, and the *"garden"* can be further partitioned into *"flower", "grass"* etc. By the virtues of this kind of hierarchy concepts, multi-level image annotating has been done by Fan et al. [Fan, et al. 2004a, Fan, et al. 2004b] and [Yuli, et al. 2006a].

[Yuli, et al. 2006a] organize keywords into different level of semantics in a hierarchical structure. At the lowest level are those concepts which can be represented by salient objects. The individual detectors of these salient objects are trained separately. Since the variation in the appearance of each salient object is relatively small, the collection of salient object detectors can achieve high classification accuracy. At the upper level is the atomic semantic image concept. They are detected in a probabilistic way by a Bayesian framework considering their dependency on the salient objects. The Figure 2.9 shows the diagrammatic representation of the keywords into different level of semantics in a hierarchical structure.

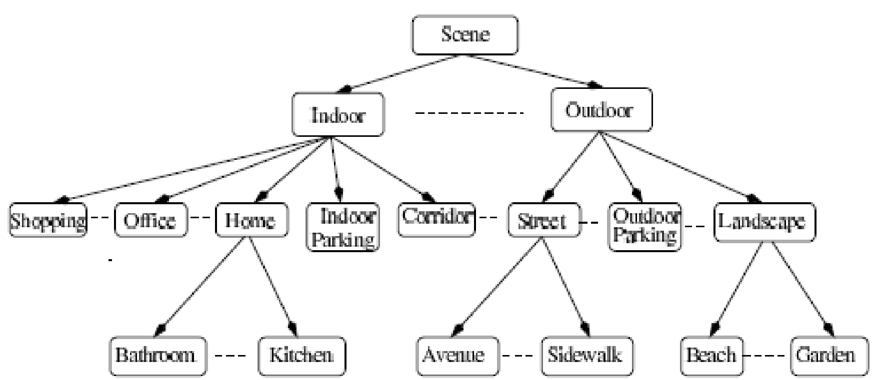


Figure 2.9: An example part of the concept ontology used by [Yuli, et al. 2006a].

As for building the concept ontology, [Yuli, et al. 2006b], proposed a semi-supervised algorithm to learn their concept ontology from the LabelMe dataset [Russell, et al. 2005] and the WordNet [Miller, 1992]. Nevertheless such a multi-level annotation framework has only been tried on a very special domain selected natural scene images. Their performance on large scale images is not clear. Especially, the generation of the concept ontology will be difficult if there are a large number of concepts in different level of semantics.

2.4.3 Semi-Automatic Annotation

Semi-automatic annotation is manual annotation with machine extraction of some information. It depends on the user's interaction to some degree. The technical information (see automatic annotation) is added automatically from, for instance, a camera; the user can then be prompted to add additional information to the image or video. The manually added information is typically semantic information. [Anita, et al. 2004]. Semi-automatic annotation combines the efficiency of automatic annotation and the accuracy of manual annotation of images. Human interaction can also provide an initial query or feedback during the annotation [Jack. et al 2005]. [Wenyin et al. 2001], describes a semiautomatic image annotation process that is better than manual annotation in terms of efficiency and better than automatic annotation in terms of accuracy. The strategy aims to combines content-based image retrieval and user verification to achieve correct high-level metadata, i.e. to create and refine annotations by *"encouraging the user",* to give *relevance feedback*, [Lu et al. 2000] of the retrieved results. That is, let the user confirm if an annotation is correct or wrong for a given image. The conclusion they made is that images annotation percentage would increase without too much user effort. This increase would be larger if an initial amount of the images' collection, for example 10%, is manually annotated. A similar approach has been adopted by [Alipr], an image search engine that retrieves images relevant to a text–based query, or similar to an image - uploaded in real time. Each image has two links to get the most similar images to it. One of this links is *"visual similar",* which returns the most similar images based on the content of the images. The other link is *"related",* which returns the most related images based on the annotations (tags) of the images. [Ivan et al 2010] described the object-based tag propagation technique for semiautomatic image annotation. According to him, when the user marks a specific object in an image, the system performs an object duplicate detection and returns the search results with images containing similar objects. Then, the annotation of the object can be performed in two ways: (1) In the tag recommendation process, the system recommends tags associated with the object in images of the search results, among which, the user can accept some tags for the object in the given image. (2) In the tag propagation process, when the user enters his/her tag for the object, it is propagated to images in the search results.

In video techniques, [Zhu et al 2002], applied video content description ontology for video, which is *Video Description (VD), Group Description (GD), Shot Description (SD) and Frame Description (FD).* The first VD were used to store information what is the video about, GD about events, SD about the object and their actions while FD store information about the what is in the frame. [Zhu et al 2002] uses automatic video segmentation techniques for Group detection, shot detection and key frame detection, while manual annotation process are perform at all level. [Yan SONG et al. 2005] proposed a semi–automatic video annotation strategy for video semantic classification, using relevance feedback to refine the classification, and active learning process to speed up the automatic learning process of classifying videos, by labeling the most informative samples. [Yan SONG et al. 2005] exploits the global and local statistical characteristics of videos, and the temporal relationship between shots. They trained the global model on a smaller pre-labeled video dataset, while local information obtained online in the process of active learning. [Yan SONG et al 2006], proposed another semiautomatic annotation framework for home videos databases based on the active learning and semi-supervised ensemble method. [Fischer, 2008], applied the semi-automatic techniques for face recognition for a TV series. He applied methods use the state of the art face detectors to detect frontal or close to frontal faces in videos, especially at shot boundaries. Then, face trackers are employed to attach images of the same face and to extract the sequence of face within a shot. Some of the tracks are labeled manually and used as the training set. Finally, the rest of the tracks are labeled automatically based on the manually labeled set.

The most popular semiautomatic tools for multimedia are the [ESP] Game and [Peekaboom] developed for collecting information about digital content. The ESP Game [Ahn et al 2004] randomly matches two players who are not allowed to communicate with each other. They are shown the same image and asked to enter a textual label that describes it. The aim is to enter the same word as your partner in the shortest possible time. Peekaboom [Ahn et al 2006] takes the ESP Game to the next level. Unlike the ESP Game, it's asymmetrical. To start, one player is shown an image and the other sees an empty black space. The first user is given a word related to the image, and the aim is to communicate that word to the other player by revealing portions of the image. Peekaboom improves on the data collected by the ESP Game and for each object in the image determines precise location information.

In short, due to the nature of semiautomatic approaches, they are usually used for preparation of training data, but in the field of annotation for multimedia- semiautomatic annotation carries the advantages and disadvantages of manual annotation and, as we will see, it also inherits the advantages and disadvantages of automatic annotation.

2.5 Video Temporal Semantic Annotation

Shot, scene and keyframes are the key component of the video for the semantic annotation. As the video is complex in nature due to its multimodal (textual, audio and visual) nature. Only keyframes is considered to be an individual image in video domain and the annotation mechanism for the images suits for the keyframes. Shot and scene semantic analysis initiates the time dimension to the problem at hand. The time dimension supplements temporal frames, resulting in more information to aid the analysis. The section is arranged by modality. We caduceus some light on multimodality shot and scene semantic analysis and keep the debate emphasis on visual information analysis.

2.5.1 Audio Analysis

Audio analysis becomes a very significant part of the multimodal analysis task when processing sport videos, TV news, movies, and so forth. Various types of audio can populate the sound track of a multimedia document, the most common types being speech, music, and silence. [Zhang, et al. 2002] propose methods to segment audio and to classify each segment as speech, music, silence, and environment sound. A k-nearest neighbor model is used at the frame level followed by vector quantization to discriminate between speech and non-speech. A set of threshold-based rules is used in order to differentiate among music, environment sound, and silence.

In most TV programs and sport videos, sound events do not overlap, but in narratives (movies and soap operas), these events frequently occur simultaneously. To address this problem, [Akutsu, et al. 1998] present an audio-based approach to video indexing by detecting speech and music independently, even when they occur simultaneously. With a similar goal, [Naphade, et al. 2000] define a generic statistical framework based on hidden Markov models [Rabiner, 1989] in order to classify audio segments into speech, silence, music, and miscellaneous and their co-occurrences.

Another important audio analysis task is the classification of the musical genre of a particular audio segment. This can capture the type of emotion that the director wants to communicate (e.g., stress, anxiety, happiness). [Tzanetakis, et al. 2002] describe their work on categorizing music as rock, dance, pop, metal, classical, blues, country, hip-hop, reggae, or jazz (jazz and classical music had more subcategories).

2.5.2 Visual Analysis

Many of the visual video analysis approaches are grounded on heuristics that are inferred empirically. Statistical approaches are more common when considering multimodal analysis. Most of the following state-of-the-art explores the temporal evolution of features to semantically analyze video content (e.g., shot classification, logical units, etc.). Video visual analysis algorithms are of two types: (a) heuristics-based, in which a set of threshold rules decides the content class, and (b) statistical algorithms.

Heuristic methods trust on deterministic rules that were defined in some empirical way. These methods monitor histograms, and events are detected if the histogram triggers a given rule (usually a threshold). They are particularly adequate for sport videos because broadcast TV follows a set of video production rules that result in well-defined semantic structures that ease the analysis of the sports videos. Several papers have been published on sports video analysis, such as football, basketball and tennis, in order to detect semantic events and to semantically classify each shot [Sezan, et al. 2003; Hwang, et al. 2003; Tan, et al. 2000]. Other heuristic methods deploy color histograms, shot duration, and shot sequences to automatically analyze various types of sports such as football [Ekin, et al. 2003] and American football [Sezan, et al. 2003].

The statistical methods reviewed formerly can be applied to the visual analysis of video content with the advantage that shapes obtained by segmentation are more accurate due to the time dimension. Also, exploring several key-frames of the same shot and then uniting the results facilitate the identification of semantic entities in a given shot.

[Hwang, et al. 2003] statistical framework tracks objects within a given shot with a dynamic Bayesian network and classifies that shot from a coarse-grain to a fine-grain level. At the course-grain level, a key-frame is extracted from a shot every 0.5 seconds. From these key-frames, motion and global features are extracted, and their temporal evolution is modeled with a hierarchical hidden Markov model (HHMM). Individual HHMMs (a single-class model approach) capture a given semantic shot category. At the fine-grain level analysis, [Hwang, et al. 2003] employ object recognition and tracking techniques. After the coarse-grain level analysis, segmentation is performed on the shots to extract visual objects. Then, invariant points are detected in each shape to track the object movement. These points are fed to a dynamic Bayesian network to model detailed events occurring within the shot (e.g., human body movements in a golf game).

2.5.3 Multimodal Analysis

In the previous analysis, the audio and visual modalities were considered independently in order to detect semantic entities. These semantic entities are represented in various modalities, capturing different aspects of that same reality. Those modalities contain co-occurring patterns that are synchronized in a given way because they represent the same reality. Thus, synchronization and the strategy to combine the multimodal patterns is the key issue in multimodal analysis.

Sports video analysis can be greatly improved with multimodal features; for example, the level of excitement expressed by the crowd noise can be a strong indicator of certain events (foul, goal, goal miss, etc). [Leonardi, et al. 2004] take this into account when designing a multimodal algorithm to detect goals in football videos. A set of visual features from each shot is fed to a Markov chain in order to evaluate their temporal evolution from one shot to the next. The Markov chain has two states that correspond to the goal state and to the nongoal state. The visual analysis returns the positive pair shots, and the shot audio loudness is the criterion to rank the pair shots. Thus, the two modalities never are combined but are used sequentially. Results show that audio and visual modalities together improve the average precision when compared only to the audio case [Leonardi, et al. 2004].

In TV news videos, text is the fundamental modality with the most important information. [Westerveld, et al. 2003] build on their previous work described previously to analyze the visual part and to add text provided by an Automatic Speech Recognition (ASR) system. The authors further propose a visual dynamic model to capture the visual temporal characteristics. This model is based on the Gaussian mixture model estimated from the DCT blocks of the frames around each key-frame in the range of 0.5 seconds. In this way, the most significant moving regions are represented by this model with an evident applicability to object tracking.

[Naphade, et al. 2001] characterize single-modal concepts (e.g., indoor/outdoor, forest, sky, water) and multimodal concepts (e.g., explosions, rocket launches) with

Bayesian networks. The visual part is segmented into shots [Naphade, et al. 1998], and from each key-frame, a set of low-level features is extracted (color, texture, blobs, and motion). These features then are used to estimate a Gaussian mixture model of multimedia concepts at region level and then at frame level. The audio part is analyzed with the authors' algorithm described previously [Naphade, et al. 2000]. The outputs of these classifiers are then combined in a Bayesian network in order to improve concept detection. Their experiments show that the Bayesian network improves the detection performance over individual classifiers. IBM's research by [Adams, et al. 2003] extend the work of [Naphade, et al. 2001] by including text from Automatic Speech Recognition as a third modality and by using Support Vector Machines to combine the classifiers' outputs. The comparison of these two combination strategies showed that SVMs (audio, visual, and text) and Bayesian networks (audio and visual) perform equally well. However, since in the latter case, speech information was ignored, one might expect that Bayesian networks can, in fact, perform better.

The approach by [Snoek, et al. 2005] is unique in the way synchronization and time relations between various patterns are modeled explicitly. They propose a multimedia semantic analysis framework based on [Allen, 1983] temporal interval relations. Allen showed that in order to maintain temporal knowledge about any two events, only a small set of relations is needed to represent their temporal relations. These relations, now applied to audio and visual patterns, are the following: precedes, meets, overlaps, starts, during, finishes, equals, and no relation. The framework can include context and synchronization of heterogeneous information sources involved in multimodal analysis. Initially, the optimal pattern configuration of temporal relations of a given event is learned from training data by a standard statistical method (maximum entropy, decision trees, and SVMs). New data are classified with the learned model. The authors evaluate the event detection on a soccer video (goal, penalty, yellow card, red card and substitution) and TV news (reporting anchor, monologue, split-view and weather report). The differences among the various classifiers (maximum entropy, decision trees, and SVMs) appear to be not statistically significant.

2.6 Annotation Using Ontology and Knowledgebase

Utilization of the semantic relationships among concepts is recently receiving a large consideration from the scientific community, since it can ameliorate the detection accuracy of concepts and obtain a richer semantic annotation of a multimedia. To this end, ontologies are expected to enhance the capability of computer systems to automatically detect even complex concepts and events from visual data with higher reliability. Ontologies consist of concepts, concept properties, and relationships between concepts. They organize semantic heterogeneity of information, using a formal representation, and provide a common vocabulary that encodes semantics and supports reasoning.

In the last years many researches have exploited ontologies to perform semantic annotation and retrieval from digital libraries. Ontologies useful for semantic annotation of multimedia are those defined by the Dublin Core Metadata. Among the recent works that follow this approach, [Snoek et al. 2007] defined *"semantically enriched detectors"* by linking a general-purpose ontology (obtained from WordNet) to a set of detectors (with several hundreds of concepts), obtaining an improvement with respect to TRECVid 2005 classification results, TV Anytime - they have defined standardized metadata vocabularies - and the LSCOM initiative [Naphade, et al 2006] - that has created a specialized vocabulary for news video. In these cases, ontologies include a set of linguistic terms with their associated definitions that formally describe the application domain, through concepts, concept properties and relations, according to some particular view.

Other ontologies provide structural and content-based description of multimedia data, similarly to the MPEG-7 standard. Garcia and Celma [15] have produced an OWL-Full ontology obtained through an automatic translation of MPEG-7; this approach has the limitation that computational complexity and decidability of reasoning is not guaranteed. [Garcia, et al. 2005] have manually developed an OWL-DL ontology that captures the full MPEG-7 Multimedia Description Schema (MDS) and the parts of the

MPEG-7 video and audio schemas that are required for the complete representation of MDS. In [Arndt, et al. 2007] an OWL-DL ontology, designed to provide a high degree of axiomatization, ensuring interoperability through machine accessible semantics, and extensibility has been proposed. This ontology comprises parts of MPEG-7 descriptors such as visual low-level, spatiotemporal decomposition and media information descriptors.

Many researchers have proposed integrated systems where the ontology provides the conceptual view of the domain at the schema level, and appropriate classifiers play the role of observers of the real world sources and classify an observed entity or event in a concept of the ontology. Classifiers have the responsibility of implementing invariance with respect to several conditions that may change the appearance of entities, such as changes in illumination, geometric perspective, occlusion, etc. Once the observations are classified, the ontology is exploited to provide an organized semantic annotation and establishing links between concepts. [Ebadollahi, et al. 2006] performed detection of events of the LSCOM ontology. Events were viewed as stochastic temporal processes in the semantic concept space and their pattern was modeled as the collection of the confidences about the elementary concepts associated with the event, computed by the detectors. [Snoek et al. 2007] proposed a method to perform video annotation with the MediaMill 101 concept lexicon. In this work machine learning technique trains classifiers to detect high-level concepts from low-level features, while WordNet is used to derive high-level concepts relations in order to enhance the annotation performances. [Zha, et al. 2007] have defined ontology to provide some structure to the LSCOM-lite lexicon, using pairwise correlations between concepts and hierarchical relationships, to refine concept detection of SVM classifiers. [Hauptmann, et al. 2007] proposed a framework to learn relationships between concepts by analyzing the co-occurrences between concepts, so as to reinforce the detection made by the classifiers. A methodology for the analysis of low-level features and semantic properties of three at concepts lexicons has been recently presented in [Koskela, et al. 2007] by Koskela, Smeaton et al., showing that modeling inter-concept relations can provide a promising resource for semantic analysis of multimedia data.

Other approaches have directly included in the ontology an explicit representation of the visual knowledge, to perform reasoning not only at the schema level but also at the data level. [Bloehdorn, et al. 2005], defined a Visual Descriptors ontology, a Multimedia Structure ontology and a Domain ontology to perform video content annotation at semantic level. The Visual Descriptors ontology included concept instances represented with MPEG-7 visual descriptors. [Dasiopoulou, et al. 2005] have included in the ontology instances of visual objects. They have used as descriptors qualitative attributes of perceptual properties like color homogeneity, low-level perceptual features like components distribution, and spatial relations. Semantic concepts have been derived from color clustering and reasoning. [Maillot, et al. 2008] have proposed a visual concept ontology that includes texture, color and spatial concepts and relations for object categorization. A set of classifiers for the recognition of visual concepts is trained using features extracted from a set of manually annotated and segmented samples.

In the attempt of having richer annotations, other authors have explored the usage of reasoning over multimedia ontologies. In this case spatio-temporal relationships between concept occurrences are analyzed so as to distinguish between scenes and events and provide more precise and comprehensive descriptions. [Neumann, et al. 2006] have proposed a framework for scene interpretation using Description Logic reasoning techniques over *"aggregates"*, these are composed of multiple parts and constrained by temporal and spatial relations to represent high-level concepts, such as objects conjurations, events and episodes. In [Espinosa, et al. 2007] manually annotated regions of images are used as visual representations of concepts, and relations between concept instances are obtained automatically. Inference from observation to explanation (abduction) is then used to check, among detected entities, relations and constraints that lead to consistent interpretation of image content. [Leslie, et al. 2007] have employed a two-level ontology of artistic concepts that includes visual concepts such as color and brushwork in the first level, and artist name, painting style and art period for the high-level concepts of the second level. A transductive inference framework has been used to annotate and disambiguate high-level concepts. In [Dasiopoulou, et al. 2008] automatically segmented image regions are modeled through low-level visual descriptors and associated to semantic concepts using manually labeled regions as training set. Context information is exploited to reduce annotation ambiguities. The labeled images are transformed into a constraint satisfaction problem (CSP) that can be solved using constraint reasoning techniques.

Several authors have exploited the ontology schema using rule-based reasoning over objects and events. [Snoek, et al. 2005] performed annotation of sport highlights using rules that exploited face detection results, superimposed captions, teletext and excited speech recognition, and Allen's logic to model temporal relations between the concepts in the ontology. [Francois, et al. 2005] defined a special formal language to define ontologies of events and used Allen's logic to model the relations between the temporal intervals of elementary concepts, so as to be able to assess complex events in video surveillance. [Hollink, et al. 2005] defined a set of rules in SWRL (Semantic Web Rule Language) to perform semi-automatic annotation of images of pancreatic cells. [Bai, et al. 2007] defined a soccer ontology and applied temporal reasoning with temporal description logic to perform event annotation in soccer videos. All these methods have defined rules that are created by human experts; thus, these approaches are not practical for the definition of a large set of rules.

To overcome this problem some researchers have studied techniques to learn automatically a set of rules. [Dorado, et al. 2004] performed video annotation based on learned rules that infer high-level concepts from low-level features using decision tree technique. [Shyu, et al. 2008] proposed a method to annotate rare events and concepts based on set of rules that use low-level and middle-level features. A decision tree algorithm is applied to the rule learning process. Moreover they addressed the imbalance problem of positive and negative examples in the case of rare event/concept using data mining techniques. [Liu et al. 2008] proposed a method to enhance accuracy of semantic concepts detection, using association mining techniques to imply the presence of a concept from the co-occurrence of other high-level concepts. None of these three works is based on ontologies and the type of rules that can be learned with these approaches cannot be directly applied to an ontology-based framework. Moreover, these methods that learn a set of rules by exploiting decision tree algorithms and low-level features, or simple junctions of high-level concepts, are not enough expressive to describe complex concepts and in particular events.

On the hand, the uses of knowledgebases also play an important role in the high level concept extraction. [Tansley, 2000] introduces a multimedia thesaurus in which media content is associated with appropriate concepts in a semantic layer composed by a network of concepts and their relations. The process of building the semantic layer uses Latent Semantic Indexing to connect images to their corresponding concepts, and a measure of each correspondence (image concept) is taken from this process. After that, unlabeled images (test images) are annotated by comparing them with the training images using a *k*-nearest-neighbor classifier. Since the concepts' interdependences are represented in the semantic layer, the concepts' probability computed by the classifier are modified by the others concepts.

Other researcher have investigated not only the statistical interdependence of context and objects but also have used other knowledge that is not existing in multimedia data, which humans use to comprehend (or predict) new data. [Srikanth, et al. 2005] incorporated linguistic knowledge from WordNet [Miller, 1992] in order to deduce a hierarchy of terms from the annotations. They generate a visual vocabulary based on the semantics of the annotation words and their hierarchical organization in the WordNet ontology.

[Benitez, et al. 2002] and [Benitez, 2005] took this idea further and suggested media ontology (MediaNet) to help to discover, summarize, and measure knowledge from annotated images in the form of image clusters, word senses, and relationships among them. MediaNet, a Bayesian network-based multimedia knowledge representation framework, is composed by a network of concepts, their relations, and media exemplifying concepts and relationships. The MediaNet integrates classifiers in order to discover statistical relationships among concepts. WordNet is used to process image annotations by stripping out unnecessary information. The summarization process implements a series of strategies to improve the images' description qualities, for example using WordNet and image clusters to disambiguate annotation terms (images in the same clusters tend to have similar textual descriptions). [Benitez, 2005] also proposes a set of measures to evaluate the knowledge consistency, completeness, and conciseness.

[Tansley, 2000] used a network at the concept level, and [Benitez, 2005] used the MediaNet network to capture the relations at both concept and feature levels. In addition, [Benitez, 2005] utilized WordNet, which captures human knowledge that is not entirely present in multimedia data.

2.7 Refining Schemes for Multimedia Annotation

Despite continuous efforts in designing new algorithms for image annotation, the performance of state-of-the-art image annotation systems are still far from satisfactory. It would be advantageous to develop approaches that could refine the annotations which have been generated by the existing annotation algorithms. One benefit of such refinement schemes is that we can incorporate additional information which could not easily be incorporated into the annotation algorithm itself.

The pioneering work of [Yohan, et al. 2005] was the first attempt at refining image annotations. They proposed to use WordNet [Miller, 1992] to calculate the relatedness between a pair of words and used the relatedness score to prune irrelevant words given a candidate set of keywords. Their pruning scheme is heuristic, i.e. given a candidate set of words, they calculate the pairwise semantic relatedness between one word to all of the rest in the candidate keywords. The words with the least semantic relatedness to all the other words are removed from the candidate annotations. In computing the semantic relatedness they combine several different measures of semantic relatedness previously proposed on WordNet [Miller, 1992]. These include the [Resnik, 1995], the [Jiang et al. 1997], the [Lin, 1997], the [Leacock, et al. 1996] and the [Banerjee, et al. 2003]. [Liu, et al. 2006] describe an annotation refinement approach which is similar to [Jin, et al 2004]. The major difference is that their computation of semantic relatedness is a weighted summation of two measures. The first one is the JNC measure [Jiang, et al. 1997], obtained from the WordNet [Miller, 1992] and the second one is derived from the empirical co-occurrence statistics on the training data.

In a related work, [Datta, et al. 2006] proposed the combination of three factors to refine image annotation: a) the word salience, b) the word frequency, and c) the word congruity. The word salience, in their context, refers to the occurring frequency of a word in a text corpus. The word frequency refers to the degree of certainty given by the annotation algorithm. The word congruity refers to the pairwise word relatedness. The overall word relatedness is a weighed summation of the above three factors. The difference between this to that given in [Yohan, et al. 2005] and [Liu, et al. 2006] is that they take into account the uncertainty of an annotating word given by the annotation process.

[Wang, et al. 2006] also considered the uncertainty of keywords given by the annotation process, but modeled the refining process as a random walk with restart (RWR) [Tong, et al. 2006] on a graph. In this graph, each node represents a candidate word. The probability of a word being given by the annotation process is viewed as the probability that the corresponding node will stay with itself and not walk to another node. The semantic relatedness between two words is represented as the probability of that random walks move from one node to another. Given this graph model, the refined annotation probability of a keyword is viewed as the steady probability of a random walk reaching the corresponding node. In their later work [Wang, et al. 2007] they modeled the annotation refining process as a Markov process. The annotated probability is modeled as the Markovian chain and the refined annotation is given by the steady probability of the chain providing a transition matrix. A major difference between this approach and the previous one in [Wang, et al. 2006] is that the transition matrix of the Markovian chain is dynamically constructed for each test image. This transition matrix, called the query biased transition matrix, takes into account not only the word relatedness and the empirical co-occurrence statistics on the training data, but also the visual similarity between the test image and those images annotated by both of the two words in consideration.

[Altadmri, et al. 2009] put forward a framework for video annotation enhancing and validation using WordNet and ConceptNet. [Altadmri, et al. 2009] enhance the existing annotation by adjoining synonym set with each term and then validate each term using ConceptNet *"capableOf", "usedFor"* and *"locationAt".* The only curb of this approach is that [Altadmri et al. 2009] does not care about the noisy keywords generated around during annotation process. For enlightening annotation, [Yohan et al. 2009] bring up the innovative approach using semantic similarity measure among annotated keywords. [Yohan et al. 2009]Detected irrelevant keywords among candidate annotated keywords by uniting evidence-rule based on semantic similarity in WordNet (TMHD model). For instance, if an image has been annotated with *'sky', 'water', 'mountain', 'door'* by TM model, TMHD model computes the semantic similarity of one word ([Yohan et al. 2009] called ('*semantic dominance'*) over all other candidate words (e.g., *'sky'* with other keywords such as *'water'*, *'mountain'* and *'door'*). TMHD model combined semantic dominance score from three different semantic similarity measurements (JNC, LIN, BNP) and keep only strong candidate annotation keywords whose scores are above the threshold. This approach diminishes the annotation diversity and hence decreases in the retrieval degree.

2.8 Evaluation Measures

The standard process of scientific research is to evaluate hypotheses and research questions based on clear and justified standards. In the last thirty years, a large variety of different evaluation metrics have been developed to evaluate the annotations ability to correctly annotate the multimedia documents, some of which are introduced in the following.

2.8.1 Tagging Ratio

The base line metrics for the tagging validation before and after processing, the tagging ratio is the average number of labels tag per image.

$$T = \frac{\sum_{i=1}^{n}(C_i)}{N} \tag{2.1}$$

Where C_i is the number of Concepts tags with the image and N is the total number of images in the datasets respectively.

2.8.2 Enrichment Ratio

The other metric for concepts enhancement during the annotation processing is enhancement ratio, which is the ratio of tagging ratio increase before and after processing.

$$E = \frac{T_1}{T_2} \tag{2.2}$$

Where T_1 and T_2 are the tagging ratio before and after process perform on the corpus.

2.8.3 Concept Diversity

The concept diversity metric for annotations expresses the different topics or concept name exist in the dataset. It's the ratio of concept tag with the documents before and after processing.

$$CD = \frac{\sum_{i}^{n} \omega}{\sum_{i}^{n} \omega'} \tag{2.3}$$

Where ω and ω' are tag concepts before and after processing.

2.8.4 Retrieval Degree

Retrieval degree is the number of correct images retrieved with a simple concept based query. Its measure is based on the precision of the query posed on the corpus. The measures introduced in the *Cranfield II experiments* [IVA] are recall and precision. They are nowadays the de facto main evaluation metrics of IR systems.

$$Precision = \frac{\#\, relevant\ document\ retrieved}{\#\, retrieved\ documents} \tag{2.4}$$

Precision is a measure of the proportion of retrieved relevant documents. It is important in information search. Considering that users often interact with few results only, the top results in a retrieved lists are the most important ones. An alternative to evaluate these results is to measure the precision of the top-N results, P@N. P@N is the ratio between the number of relevant documents in the first N retrieved documents and N. The P@N value focuses on the quality of the top results, with a lower consideration on the quality of the recall of the system.

$$Recall = \frac{\#\, relevant\ document\ retrieved}{\#\, relevant\ docuemnt\ in\ the\ collection} \tag{2.5}$$

The recall measures the proportion of relevant documents that are retrieved in response to a given query. A high recall is important especially in copyright detection tasks. In high level semantic annotation and propagation precision and recall are defined slightly differently. There are two versions, per-image based and per-semantic description based.

2.8.5 Per-image Precision and Recall

Per-image precision and recall are calculated on the basis of a single test image taking from the corpus prepared for the high level semantic propagation. For each test image, precision is defined as the ratio of the number of semantic description that are correctly predicted to the total number of possible semantic description prediction tag with the image in the cluster set, and recall is the ratio of the number of semantic description that are correctly predicted to the number of semantic description in the cluster sets. Mathematically, they are calculated as follows

$$Per\ Image\ Recall = \frac{\#\ correclty\ semantic\ description}{\#\ semantic\ description\ of\ images\ in\ cluster\ set} \qquad (2.6)$$

$$Per\ Image\ Precision = \frac{\#\ correclty\ semantic\ description}{\#\ total\ semantic\ description\ tags\ with\ images\ in\ cluster} \qquad (2.7)$$

Per-image precision and recall values are averaged over the whole set of cluster images to generate the *mean per-image* precision and recall.

2.9 Chapter Summary

In this chapter, we surveyed the several different principles that are used in the image and video annotation. We first discussed the fundamental concepts related with the multimedia annotation, followed by multimedia annotation description standards and then the purpose of each standard. The detailed discussion about the different methods of multimedia presented in the three subsection under the head of manual, automatic and semiautomatic annotation, while the temporal annotation for video are discussed separately. To achieve more comprehensive investigation, we present the ontological and knowledgebases approaches followed by discussion related with annotation refinement scheme for multimedia annotation. At the end of the chapter we have briefly discussed the evaluation measure and metrics.

We have done a detailed survey of all the techniques used for multimedia annotation and concluded that semantic based annotation using ontological or knowledgebase approaches outperforms then the content based techniques. Keeping this in mind we have further contributed in the Semantic based annotation and refinement by proposing three main contributions which are discussed in the forthcoming chapters. The detailed discussion of the first contribution about the annotation enhancement and refinement will be found in the coming chapter 3.

Chapter 03

A Framework for Image Annotation Enhancement & Refining Using Knowledgebases

"All truths are easy to understand once they're discovered; the point is to discover them."
Galileo

Semantically enriched multimedia information is crucial for equipping the kind of multimedia search potentials that professional searchers need, while on the other side the expansion growth of multimedia (images and video) data online has the potential to encourage more erudite and vigorous models and algorithms to systematize, index, retrieve multimedia and the like corpus. On the contrary, inclusively how much data can be hitched and systematized remains a critical problem, also the semantic interpretation of multimedia is obsolete without some mechanism for understanding semantic content that is not explicitly available. However, Manual annotation is the exclusive source to overwhelming this, which is not only time consuming and costly but also lacks semantic enrichment in terms of concept diversity and concept enrichability as well.

In this chapter, we present semantically enhanced information extraction model that prune the initial tags from noisy and unusual words attached with the images by using stopwords, unification and redundancy control approaches and afterwards the purified tags are enhanced lexically and commonsensically using the knowledgebases .i.e. WordNet and ConceptNet. By doing this a lot of noises, redundant and unusual keywords are again generated, which are then filtered out by using semantic similarity as a process performs for the concept refinement. Results show that searching for an image over enhanced tags outperforms searching using the original annotated terms. We achieve good results in terms of concept enrichment ability, retrieval performance and concept diversity.

The rest of the chapter is organized as follows: In section 3.1, a brief introduction about the work is mentioned, while state-of-the-art is the part of section 3.2. A detail about the propose framework covering depth of each module with their algorithm is presented in section 3.3. Experimental work is discussed in Section 3.4, where we present how effectively our proposed framework improves the retrieval degree of the LabelMe dataset. We achieve a noticeable improvement in terms of enrichment ratio, concept diversity and retrieval degree. The chapter is finally concluded along with future work in section 3.5.

3.1 Introduction

Historically, images have been retrieved by the librarians, initially manually annotating them with one or more keywords or more specifically concepts with a single goal in mind that is to describe the image contents. For a given query, these annotations are used to retrieve appropriate images. Underlying this approach is the belief that the keywords associated with an image essentially capture the semantics of the image and any retrieval based on these keywords will, therefore, retrieve relevant images. Queries based on images visual attributes like colour, texture or shape have been widely proposed for retrieving images, but it is difficult for most of the users to use that kind of visual attributes. Most people would prefer to pose text queries and find images relevant to those queries. Keeping this today, many front line search engines like Google, Yahoo, including mobiles (e.g., Google Mobile, and Yahoo! Mobile) rely on keyword based retrieval. In many scenarios, we want to find the images related to a specific concept, i.e. "Park" or we want to find the keywords that best describe the contents of an unseen image [Duygulu, et al 2002]. Sometime the annotator (manual or automatic) goes wrong to express the semantics accurately and while sometimes it is even worse, that the user query semantic space is quite different to the ones used in the annotation describing the same semantics. That means a gap exists between users query space and an image representation space, which leads to the lower precisions and recalls of queries. The user may get an overwhelming but large percent of irrelevant images in the result sets. In fact, this is a tough problem in multimedia retrieval systems.

An effective method for solving the above problems is annotation-based image retrieval, an image collection is searched based on a textual description of the depicted content. While this advent is best-suited in situations where the desired pictorial information can be effectively illustrated by means of keywords, it demands for interpretation of the depicted contents into a textual representation (annotation), which is either done manually or by automatic means, because content-based image retrieval (CBIR) computes relevance based on the visual similarity of low-level image features such as colour histograms, textures, shapes, and spatial layout had shown their limitation. However, the complication is that visual similarity is not semantic similarity. There is a gap between low-level visual features and semantic meanings. The so-called semantic gap, which is the major problem and that needs to be solved for most of the CBIR approaches. For instance, a CBIR system may answer a query request for 'red rose' with an image of a 'red ball'. If we provide annotation of images with keywords, then a typical way to bring out an image data repository is to create a keyword-based query interface for an image database. Images are retrieved if they contain (some combination of the) keywords specified by the user. To achieve all these goals several statistical models have been suggested. For example, the translation model (TM) [Duygulu, et al 2002], the cross-media relevance model (CMRM) [Jeon, et al 2003] and a continuous relevance model (CRM) [Lavrenko, et al 2004] can determine a set of keywords that describe visual objects /regions, which appear in an image. However, whatever model we employ the current annotation accuracy is comparatively low due to the existence of the image of representation with fewer keywords that producing less semantic space. Therefore, it is quite difficult to get a meaningful understanding of images in this manner. Similarly, the multitudes of ways in which the same concept can be described pose no trouble to humans but are a particular obstacle to successful information retrieval, (IR). Bates points out in [Bate et al. 1986], "the probability of two persons using the same term in describing the same thing is less than 20%", and [Furnas et al. 1987] found that “the probability of two subjects picking the same term for a given entity ranged from 7% to 18%”. It is thus not surprising that only limited success is achievable with traditional IR approaches where information is viewed in terms of context independent single index and query terms matched as strings.

The intention of this paper is to facilitate the steps to achieve a semantic understanding of images, while the semantic meaning of images will be expressed by a set of keywords or concepts tagged with the images. We are proposing a framework for Annotation Enhancement and Refinement using Knowledgebases. This approach has three important impacts. First, almost all the previous approaches use only the base annotation either done manually or by automatic means. Taking idea from query expansion, we use annotation expansion by using lexical and commonsensical knowledgebases. Secondly, the noisy and unusual keywords are controlled by utilizing the stopwords and unification mechanism, while redundant instances of keywords take to one instance. Third, by the help of semantic similarity using WordNet, most of the irrelevant words are controlled and discarded from the data sets. This benefits not only the user to achieve a high level of accuracy for their worst queries but also provide an opportunity for the images with fewer concepts tag. Our proposed framework has been employed on the LabelMe data set for images, which is the open source dataset available for research. From the experiments, we achieve significant increases in terms of retrieval degree, annotation enrichment ratio and concept diversity.

3.2 State-of-the-Art

We can classify most of the existing automatic image annotation algorithms into two categories. First, they formulate automatic image annotation to classification problems with considering keyword (concept) as a unique class of the classifier, which are SVM classifier [Gao, et al 2006, Cusano et al 2004 and Yang, et al 2006] Gaussian Mixture Hierarchical Model [Carneiro, et al 2005a], [Carneiro, et al 2005b], Bayes Point Machines [Chang, et al 2003], 2-dimensional Multi-resolution Hidden Markov Model [Li, et al 2003] and so on. Second, many statistical models have been published for image annotation. [Mori et al. 1999] used a co-occurrence model, which estimates the correct probability by counting the co-occurrence of words with image objects. [Wei-Chao Lin et al. 2010] uses of the Information Gain (IG) and AdaBoost learning algorithms for noise and outlier information filtering in the system training stage with the hope that improve the performance of image classification. [Duygulu et al. 2002], strived to map keywords to individual image objects. Both dealt with keywords as one language and blob-tokens as another language, allowing the image annotation problem to be observed as translation between two languages. Using some classic machine translation models, they annotated a test set of images based on a large number of annotated training images. Based on translation model, [Pan et al. 2004] have put forward various methods to discover correlations between image features and keywords. They have applied correlation and cosine methods and introduced SVD as well, but the work is still based on a translation model with the seizure that all features are equally important and no knowledgebase (KB) has been used. The problem of the translation model is that frequent keywords are associated with too many different image segments but infrequent keywords have little chance of appearing in the annotation. To figure out this problem, [F. Kang et al. 2004] suggested two modified translation models for automatic image annotation and achieve better results [Kang, et al 2004]. [Jeon et al 2003] introduce cross media relevance models (CMRM) where the joint distribution of blobs and words is learned from a training set of annotated images. Unlike translation model, CMRM expects there are many to many correlations between keywords and blob tokens rather than one to one. Therefore, CMRM genuinely takes into account context facts. Furthermore, [Lavrenko et al, 2004] propose a continuous relevance model by separating an image into a fixed number of grids and avoiding segmentation and clustering issues that are observed in previous models. [Guangyu Zhu et al.

2010] applied decomposition techniques on the user provided tag matrix into a low-rank refined matrix and a sparse error matrix and targeting the optimality measure with low-rank, content consistency, tag correlation, error sparsity. However, in all of this work annotation contains many noisy keywords and there is no attempt to extend this "limit" of automatic image annotation problem.

[Amjad et al 2009] put forward a framework for video annotation enhancing and validation using WordNet and ConceptNet. [Amjad et al 2009] enhance the existing annotation by adjoining synonym set with each term and then validate each term using ConceptNet *"capableOf"*, *"usedFor"* and *"locationAt"* relations. The only curb of this approach is that, [Amjad et al 2009], does not care about the noisy keywords generated around during annotation process. For enlightening annotation, [Barrat et al. 2010] propose probabilistic graphical model to represent weakly annotated images, where they classify images and extend existing annotation to new images by considering semantic relation between keywords. [Yohan et al. 2005], bring up the innovative approach using semantic similarity measure among annotated keywords. [Yohan et al. 2005], Detected irrelevant keywords among candidate annotated keywords by uniting evidence-rule based on semantic similarity in WordNet by the help of Translational Model based Hybrid Dempster (TMHD) model. For instance, if an image has been annotated with *'sky', 'water', 'mountain', 'door'* by TM model, TMHD model computes the semantic similarity of one word [Yohan et al. 2005] called *'semantic dominance'*) over all other candidate words (e.g., *'sky'* with other keywords such as *'water', 'mountain'* and *'door'*). TMHD model combined semantic dominance score from three different semantic similarity measurements (JNC, LIN, BNP) and keep only strong candidate annotation keywords whose scores are above the threshold. This approach reduces the annotation diversity and hence decreases in the retrieval degree.

To overwhelm the inadequacy of [Amjad et al 2009, Barrat et al. 2010 and Yohan et al. 2005], we are proposing a newfangled framework for annotation enhancement and refinement that will expand lexically and commonsensically the annotation by utilizing the well-known knowledgebases. The main theme of the proposed framework is to take annotated datasets (either generated manually or by automatic means) and perform the data filtration process on that, which includes redundancy control, stopwords process and unification of the different forms of words. Next to expand the terms lexically and commonsensically via well-known knowledgebases i.e. WordNet and ConceptNet, while this process generates a set of keywords where some of the terms are related whilst several are irrelevant and that's need to be remove. In order to remove irrelevant keywords, we applied semantic similarity threshold between original keyword and that of generated keywords by utilizing the WordNet and terms equal or above the threshold are retain in the list, while others are discarded. The output of this framework is in the form of XML document for each image based on the [LabelMe] annotation structure that can be used for further processing and portability. Keeping flexible nature of this framework, so that not only can easily be plugging to any image's corpus, but also can be integrated with any other knowledgebases or domain ontologies. Moreover, the latest release of the WordNet and ConceptNet can be accommodated by only updating their API's.

3.3 Proposed Framework

The relative success of the approach debated in the literature review raises the question of whether we need images with additionally detail annotation (which is more laborious intensive to acquire than just captions). We are arguing that detailed annotation is necessary for several reasons. First, labelled data is essential for a quantitatively measure performance of different methods (i.e. object detection). Secondly, the current segmentation and interest point techniques are not capable of discovering the outlines/shapes of many object categories, which are often small or unclear in natural images. Thirdly, the annotation should be expanded and refine to fill the gap between the user query space and annotation space. As far as concern the *"semantic gap"* between concept (keyword) and low-level visual feature values. The way of image understanding for human is not depended on low-level visual feature, but human would like to rely on their knowledge which came from previous personal experiences. To bridge the semantic gap, we should try to reflect the way of human perception for image understanding. WordNet and ConceptNet, which are quite famous lexical and commonsensical knowledgebases for information research area, can be useful resources for simulating the human perceptional semantic knowledge. In text retrieval, the techniques among the others like semantic similarity got quite popularity in solving the problems of query expansion, word sense disambiguity and topic classification.

Based on the realities and problem facing by the research community using multimedia annotated datasets for search and retrieval, the proposed framework is presented in Figure 3.1., we adopt the modular approach, where each of the module is dependent on the output of the other.

Let $L = \{t_1, t_2, \dots, t_n\} = \sum_{i=1}^{n} t_i$ be the list of the label tag per image, then the corpus is,

$$C = \{x_1, x_2, \dots, x_n\} = \bigcup_{j=1}^{m} x_j \tag{3.1}$$

Where C is the corpus of images dataset representing list of the annotated images, where x represent individual image. By combining both of the equations, the equation 3.1 become

$$C = \bigcup_{j=1}^{m} \left(\bigcup_{i=1}^{n} t_i\right)_j \tag{3.2}$$

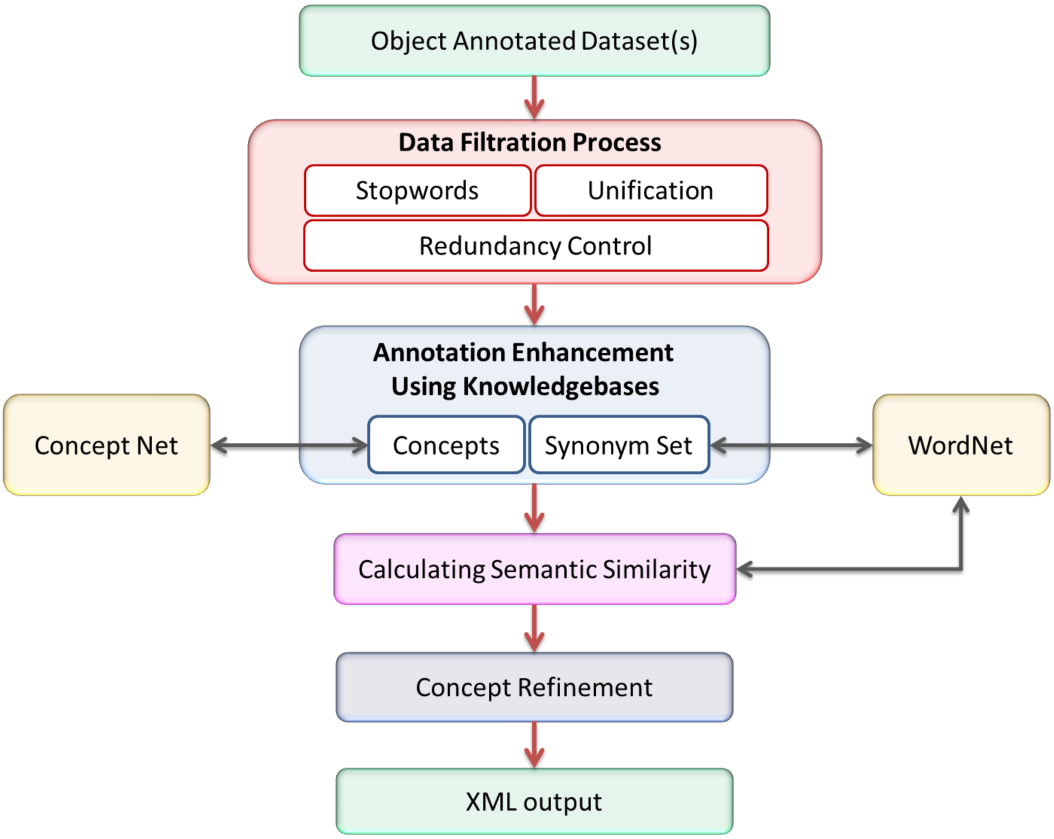


Figure 3.1: A framework for annotation expansion & refinement using lexical and commonsensical knowledgebases

3.3.1 Data Filtration Process (DFP)

The degree of freedom while using the LabelMe online annotation tool makes the users comfortable on one side, but it gains complexity in term of usability of datasets for research. Hidden problems like redundancy, irrelevant and unusual keywords are continuously generated during the annotation. The effective way of minimizing the risk during the DFP, we extend the DFP to further sub-modules, i.e. stopwords, unification and redundancy control. The output of DFP is in XML format that contain the purified form of data for the source image.

3.3.1.1 Stopwords Module

Stopping is the process of removing frequently occurring terms from indexes and queries (Witten et al., 1999). The reason for this process is that terms that transpire in most annotated documents are not very useful for recognizing relevant documents. For example, the word *"the"* occurs in most documents. If *"the"* was used as part of an annotated document or of a query, it would not have a significant impact on the answer set, if any at all. In case of LabelMe datasets, stopwords include*"az0003", "ghkdf65we", "oi45nelfds"* are totally worthless and hence no need to be further process. Stopping has two main advantages: first, the index size is reduced by a small percentage, resulting in decreased storage requirements. Second, during query evaluation, the inverted lists for stopwords, which are usually longer than average, need not be processed, which can lead to a considerable time saving. In addition, the stop words are a word that does not carry meaning in natural language. Generally, semantics of nouns is easier to identify and to grasp since nouns have meaning by themselves. Therefore, articles, prepositions, and conjunctions are natural candidates for a list of stopwords. Since stopwords elimination also provides for compression of the indexing structure, the list of stopwords might be extended to include words other than articles, prepositions, and conjunctions. For instance, some words like *"az0003", "ghkdf65we", "oi45nelfds"* could be treated as stopwords. During the process of stopwords, a list of stopwords are prepared and properly updated during the DFP. Let 'x' represent the list of words present in the annotated document A and that needs to be pass from the stopwords module to remove the unusual words, the mathematical form is as,

$$X = \{x | x \in A\} \quad (3.3)$$

Where X is the total number of documents in the corpus, let y represent the list of stopwords that need to be remove from each of the document of the corpus X, then the mathematical form of the stopwords list are,

$$X' = \{x, y | x \in A \land y \notin A\} \quad (3.4)$$

3.3.1.2 Unification Module

Unification is the process of converting the complex words into simple, the purpose of this module is two folds. First, to convert the unusual keywords into meaningful keywords, secondly conversions of keywords to the base form. As per the requirements of unification module, we have divided it into further two sub-sections, .i.e.

i. The unusual keywords, that have the object names along with some other data or information and jointly their meaning is purposeless. For example, the words like *"personsitting", "personoccluded", "personstanding"* and *"personwalking"* that include the object name and other information as well, the keywords like these needs to be unified. We have built a repository where these types of keywords are recorded throughout the corpus and then pass through the process of unification to get actual form of the keywords.

ii. The other issue is related with exact form of the keywords, for instance, words like *"fishing", "fished", "fish"* and *"fisher"* are mostly used, but from annotation point of view, we are interested only in their base form i.e. *"fish"*. The common way of controlling such inconveniences is by applying stemmer or lemmatizer. Next, we have discussed both approaches,

a) Stemming

A user often stipulates a query but only a divergent of this word is present in a relevant document. Plurals, gerund forms, and past tense suffixes are typical examples of syntactical variations, which prevent a best match between a query word and a corresponding document word. This complication can be partially overcome with the substitution of the words by their relevant stems.

A stem is the fraction of a word, which is left after the removal of its affixes (i.e., prefixes and suffixes). A typical example of a stem is the word *connects* which is the stem for the variants *connected*, *connecting*, *connection*, and *connections*. Stems are thought to be useful for improving retrieval performance because they reduce variants of the same root word to a common concept. Furthermore, stemming has the secondary effect of reducing the size of the indexing structure because the number of distinct index terms is reduced.

While the argument encouraging stemming seems sensible, there is wider debate in the literature about the benefits of stemming for retrieval performance. In fact, different studies lead to rather conflicting conclusions. [Frakes, 1998] compares eight distinct studies on the potential benefits of stemming and concludes that the results of the eight experimental studies he explored do not reach satisfactory results although he favors the usage of stemming. Because of these doubts, many Web search engines do not employ any stemming algorithm whatsoever.

In affix riddance, the genuine significant part is suffix removal because most variants of a word are aroused by the introduction of suffixes. While the Lovins algorithm, the Paice/Husk algorithm is well known suffix removal algorithms, the most popular one is that by Porter because its simplicity and elegance, which is trying to "normalize" the tokens and given them a standard form. It looks for prefixes or suffixes for a given token and yields token, so called stem. For example, ran → ran, running → run, cactus → cactus, cactuses → cactus, dog's → dog, communities → community, community → communiti.

A stemmer is expected to turn inflected forms of words down to some common root. But stemming usually results in a chop-off of the ends of words into the stem form which is usually not even a real word. It helps to sum up derivatives but inevitably loses the part-of-speech information which is crucial. It's not actually a stemmer's line of services to make words to a 'proper' dictionary word. For overwhelming this, we need to look at morphological/orthographic analyzers that take the responsibility of making root to a "proper" dictionary word.

b) Lemmatizer

Lemmatizer is one of the module of Montylingua [Covington, et al. 2007], is an automatic NLP tool that first tags input data with a tagger that the creator [Hugo Liu, 2004] claims exceeds the accuracy of the Transformation-based Part of Speech Tagger. The lemmatizer strips the suffixes from plurals and verbs and returns the root form of the verb or noun. Lemmatization is the procedure of deciding the lemma for a given word. So various inflected forms of a word can be investigated as a single item. It does a similar task with stemming but answer the dictionary form of a word and save the part of speech information for us and convert the diverse morphological form to the base form. We run the Lemmatization instead of Stemming on the datasets.

Some examples of the lemmatization output,

- Walks, walk, walking, walked → walk.
- striking → striking
- loves, loved → love
- are, am, is → be
- best, better → good

3.3.1.3 Redundancy Control Module

Redundancy is the most common problem exists in the LabelMe datasets, which is due to the fact of existing of too many similar objects in the image. For instance, if the image of the building is given, then window and door, etc. are the common words that's to be expected as redundant and that need to be control for two purposes, firstly to reduce the processing overhead and secondly restraining duplicity in result. We applied a *unique function* for redundancy control.

Let $L' = \{t_1, t_2, \ldots, t_n\} = \sum_{i=1}^{n} t_i'$ represent the purified list of the labels tag with the image, then equation 3.2 for the corpus become

$$C' = \bigcup_{j=1}^{m} \left(\bigcup_{i=1}^{n} t_i' \right)_j \tag{3.5}$$

The algorithmic presentation of the data filtration process is,

Propose Algorithm 3.1: Data Filtration Process

Input: $L \rightarrow \bigcup_{j=1}^{m}(\bigcup_{i=1}^{n} t_i)_j$

Output: $L_f \rightarrow \bigcup_{j=1}^{m}(\bigcup_{i=1}^{n} t_i')_j$

Method:

i → Length (L)

$L' \leftarrow MontyLongua.Lemmatization(L(i).name)$

IF (stopwords(L')) THEN continue

ELSE IF (replacewords(L')) THEN

L" ← replace(L')

$L_f(i) \leftarrow L"$

$L_f \leftarrow Unique(L_f)$ *//Redundancy Control*

3.3.2 Annotation Enhancement Using Knowledgebases

The algorithm for annotation enhancement using knowledgebases is presented. A Knowledgebase is a highly valued type of database for knowledge management, as long as the means for the computerized collection, organization, and retrieval of knowledge. Investigation in text mining domain manages to figure out sizable commonsense knowledgebases. The commonsense is the information and facts that are expected to be commonly known by ordinary people. Many applications in modern information technology utilize these knowledgebases for semantic web, document classification and multimedia annotation, search and retrieval. WordNet [Fellbaum, et al. 1998], CYC [Lenat, et al. 1995] and ConceptNet [Liu, et al. 2004] are considered to be the widest commonsense knowledge bases currently in use. In the proposed algorithm, we have utilize the functionality of WordNet and ConceptNet jointly, for simplicity, we have developed functions *WordNet.getSynset()* for synset and *ConceptNet.getConceptset()* for conceptset that automatically extracts the synset and conceptset from WordNet and ConceptNet respectively.

3.3.2.1 WordNet

WordNet [Carneiro, et al. 2005] is an electronic thesaurus that models the lexical knowledge of English language. The most facial feature of WordNet is that it arranges the lexical information in relations of word meanings instead of word forms. Particularly, in WordNet words with the same meaning are grouped into a "synset" (synonymous set), which is a matchless representation of that meaning. Consequently, there exists a many-to-many relation between words and synsets: some words have several different meanings (a phenomena known as polysemy in Natural Language Processing), and some meanings can be expressed by several different words (known as synonymy). In WordNet, a variety of semantic relations is defined between word meanings, represented as pointers between synsets.

WordNet is separated into sections of five syntactical categories: nouns, verbs, adjectives, adverbs, and function words. In our work, only the noun category is explored due to the following two reasons: (1) nouns are much more heavily used to describe images than other classes of words, and (2) the mapping between nouns and their meanings, as well as the semantic relations between nominal meanings are so complicated that the assistance from thesaurus becomes indispensable. WordNet [Miller, 1992] contains approximately 57,000 nouns organized into some 48,800 synsets. It is a lexical inheritance system in the sense that specific concepts (synsets) are defined based on generic ones by inheriting properties from them. In this way, synsets establish hierarchical structures, which drive from generic synsets at higher layers to specific ones at lower layers. The relation between a generic synset and a specific one is called Hypernym/Hyponym (or IS-A relation) in WordNet. For example, conifer is a hyponym of tree, while tree is a hypernym of conifer. Instead of having a single hierarchy, WordNet selects a set of generic synsets, such as {food}, {animal}, {substance}, and treats each of them as the root of a separate hierarchy. All the rest synsets are assigned into one of the hierarchies starting with these generic synsets. Besides the Hypernym/Hyponym relation, there are some other semantic relations such as Meronym/Holonym (MEMBER-OF), and Antonym. Some synsets and the relations between them are exemplified in Figure 3.2(a, b).

Words are arranged semantically and not alphabetically unlike most dictionaries. The potential benefit that WordNet has over other dictionaries is the assembling which has been applied to each word. Words are harmonized together to form synsets (synonym sets), which represent a single sense.

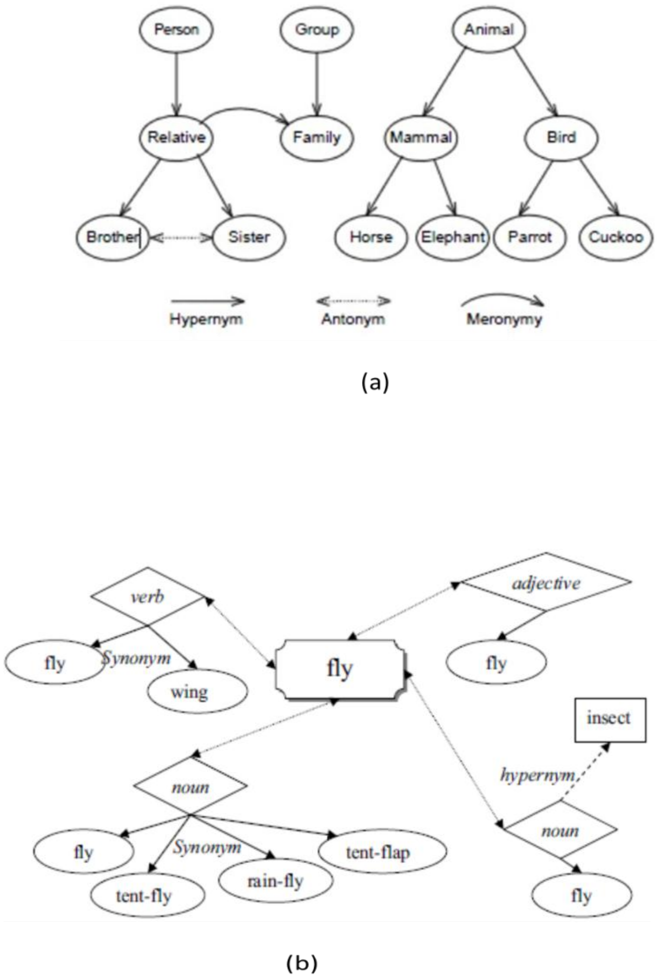


Figure 3.2: Example of synsets and semantic relations in WordNet

3.3.2.2 ConceptNet

ConceptNet [Liu, et al. 2004] is a commonsense knowledgebase. ConceptNet 2.1 also encompasses Montylingua, a natural-language-processing package. ConceptNet is written in Python but its commonsense knowledgebase is stored in text files. Unlike other knowledgebases like CYC, FrameNet and Wikipedia, ConceptNet is based more on Context and allow a computer to understand new concepts or even unknown concepts by using conceptual correlations called Knowledge-Lines. ConceptNet is at present deliberated to be the biggest commonsense knowledgebase. [Liu, et al. 2004], [Hsu, et al. 2008]. It is composed from more than 700,000 free text contributors assertions. Its nodes core structure is concepts, where each of which is a part of a sentence that expresses a meaning. ConceptNet is a very wealthy knowledgebase for several aspects: First, it includes an immense number of assertions and nodes. Second, it has a broad range of information. Finally, it has different kinds of relationships, including description parameters. Figure 3.3 presents a snapshot that includes useful relationships between concepts. In the last version of ConceptNet "ConceptNet4" each relationship has several fields expressing its score, polarity and generality. This information is automatically inferred by examining the frequency of the sentences that provoked this relationships.

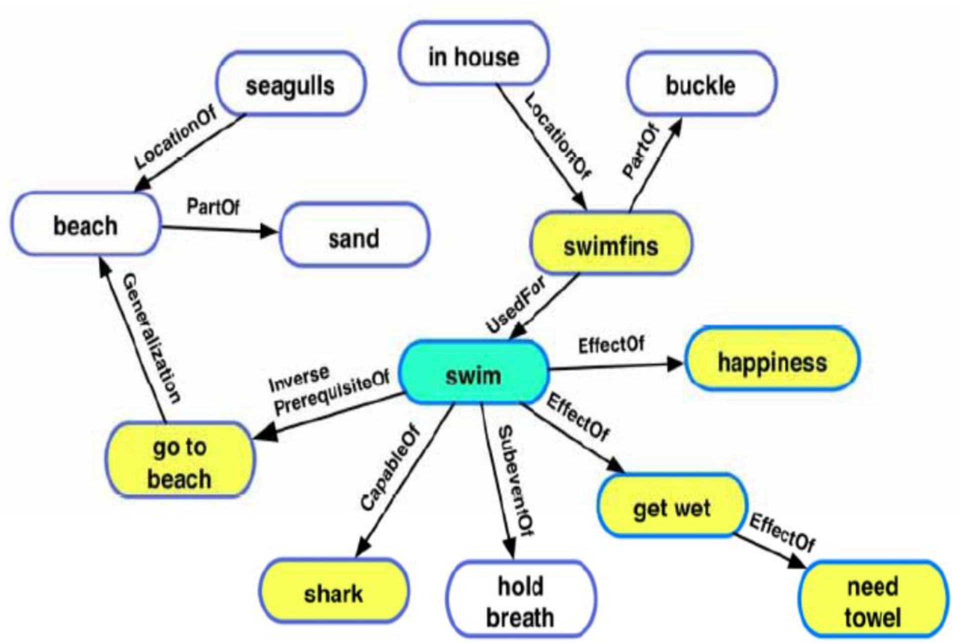


Figure 3.3: An illustration of a small section of ConceptNet

We consider the Annotation Enhancement (aE) task, where the system extends the existing annotation for each image from $x \in X$ in the purified corpus C' by using lexical and commonsensical knowledgebases, which extends the equation previous equation to,

$$C' = \cup_{j=1}^{m} (\cup_{i=1}^{n} (t', aE,))_j \quad (3.6)$$

The algorithm for annotation enhancement using lexical and commonsensical knowledgebases is presented below,

---

Propose Algorithm 3.2: Concept Expansion

Input: $L_f \rightarrow \bigcup_{j=1}^{m}(\bigcup_{i=1}^{n} t_i')_j$

Output: *L → Concept, SynSet.name, ConceptSet.name*

Method:

$i \rightarrow Length\ (L)$

    $L(i).Concept \leftarrow L_f(i)$

    // extracting and adding SynSet

    $L_s \leftarrow$ *WordNet.getSynSet(i)*

    $j \rightarrow Length\ (L_s)$

        $L(i).SynSet(j).name = L_s(j)$

    // extracting and adding SynSet

    $L_c \leftarrow$ *ConceptNet.getConceptSet(i)*

    $k \rightarrow Length\ (L_c)$

        $L(i).ConceptSet(k).name = L_c(k)$

3.3.3 Calculating Semantic Similarity

Using semantic similarity, we would like either to remove or replace noisy keywords from annotated documents and the keywords generated by proposed model. For this, we measured similarity between original keywords and each of the generated keywords. Finally, some concepts corresponding keywords discarded in which total similarity measure of an original concept with other concepts falls below a certain threshold. Following is the review of semantic similarity using knowledgebase (i.e. WordNet).

Semantic word similarity has been greatly studied, and there is numerous semantic word similarity measures commenced in the literature. Due to the subjectivity in the definition of the semantic word similarity, there is no singular way to work out the implementation of the recommended measures. The knowledge-based measures try to quantify the similarity using the information drawn from the semantic networks. Most of these measures use WordNet as the semantic network, where the semantic relations are explicitly defined that connects each of the synsets to one another. Some of these relations (hyponym, hypernym for nouns, and troponym and hypernym for verbs) constitute is-a-part-of (meronym for nouns) and is-a-kind-of (holonym) hierarchies. The similarity between two concepts and two words is not same. Since one word may have a number of senses, it can correlate to several concepts. Some of these similarity measures utilize information content (IC) which exhibits the amount of information belonging to a concept. It is described as:

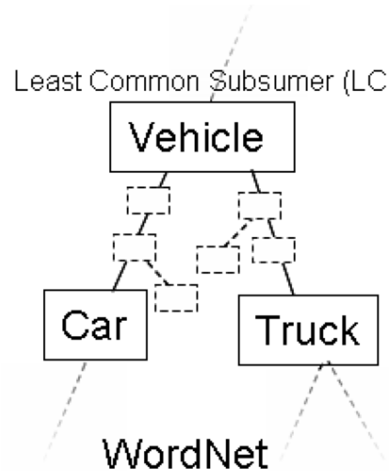


Figure 3.4: In this example LCS of the concepts car and truck is the vehicle in the given taxonomy.

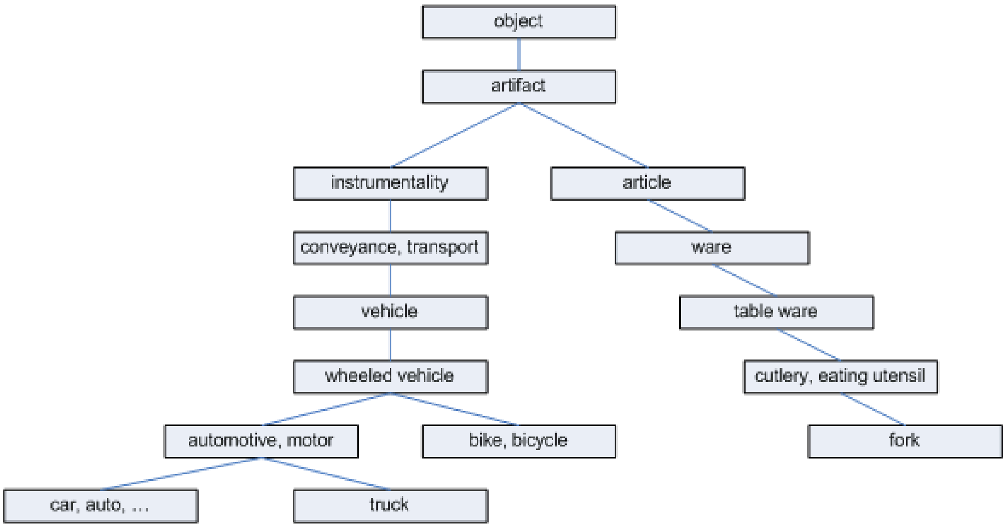


Figure 3.5: The figure [Thanh] shows an example of the hyponym taxonomy in WordNet used for path length similarity measurement, we observe that the length between car and auto is 1, car and truck is 3, car and bicycle is 4, car and fork is 12

.

In the following parts, we discuss seven distinctive knowledge-based similarity measures.

3.3.3.1 Resnik Measure (RIK)

[Resnik, et al. 1995] introduce first *Information Content (IC)* notion by relying node based approach. More, higher value of IC (Information Content) means that the concept has specified and detailed information. For example, *cable-television* has more specific information than television. RIK first uses Corpus (in our case LabelMe/Image) to get the probabilities of each concept and computed how many times the concept appear in the Corpus.

$$freq(c) = \sum_{n \in word(c)} count(n) \tag{3.7}$$

Where $\mathrm{word}(c)$ is the set of words subsumed by concept *c*. Next, the probabilities of each concept are calculated by the following relative frequency.

$$Prob(c) = \frac{freq(c)}{N} \tag{3.8}$$

Where *N* is the number of nodes. If only one root node is selected, the probability of that node will be 1. This is because root node concept subsumes every concept in WordNet. Second, RIK calculates IC of a concept by taking the negative logarithm of above mentioned probability. Finally, semantic similarity between two concepts will be calculated in the following way. First, RIK determines *Lowest Common Subsume* (*LCS*) between two concepts and then for that LCS concept IC will be determined.

$$IC(concept) = -\log Prob(concept) \tag{3.9}$$

$$sim(w_i, w_j) = max_{c_i,c_j}[sim(c_i, c_j)] \tag{3.10}$$

Note that a keyword may be associated with more than one concepts in WordNet. However, the keyword will be associated with a single concept. For example, keyword w_1 and w_2 are associated with a set of concepts c_1 and c_2 respectively. Base on that, pair wise similarity between set of concepts c_1 and c_2 are calculated and keep pair (c_1, c_2) which yields maximum value. Therefore, word similarity takes into account the maximal information content over all concepts of which both words could be an instance. RIK measure does neither consider the IC value of two concepts/ keywords, nor the distance between concepts/keywords in the WordNet. If we consider the similarity between studio and house in Figure 3.6, the LCS will be the building and its IC value will be 9.23. However, this value will be the same as the value between house and apartment. This is the weakness of RIK measure.

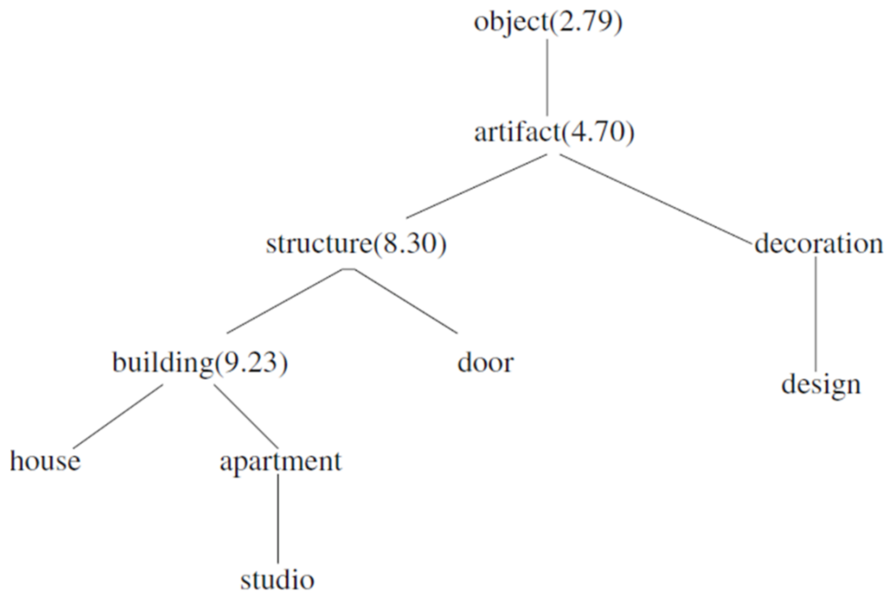


Figure 3.6: An example of information content in the WordNet [Yohan et al 2009].

3.3.3.2 Jiang and Conrath Measure (JNC)

[Jiang, et al. 1997] use the same notion of the Information Content and takes into account the distance between selected concepts. In regard to this, JNC combines node-based and edge-base approach. Let us consider the above example. Hence, the two different pair of keywords (studio and house, studio and apartment) has the same semantic similarity based on RIK measure. There is no way to discern the semantic similarity between them. However, with regard to semantic similarity between two concepts, JNC uses the IC values of these concepts along with the IC value of LCS of these two concepts. Therefore, the similarity will be different since the IC value of house and apartment are not the same. It is defined as below:

$$Sim_{jnc}(c_i, c_j) = \frac{1}{IC(c_i)+IC(c_j)-2*IC\left(LCS(c_i,c_j)\right)} \tag{3.11}$$

3.3.3.3 Lin's Measure (LIN)

The key idea in this measure is to find the maximum information shared by both concepts and normalize it. Lin's similarity [Lin, et al. 1998] is measured as the information content of LCS, which can be seen as a lower bound of the shared information between two concepts, and then normalized with the sum of information contents of both concepts. The formulation is as below:

$$Sim_{lin}(c_i, c_j) = \frac{2 \times IC(LCS(c_i,c_j)}{IC(c_i) + IC(c_j)} \tag{3.12}$$

3.3.3.4 Leacock & Chodorow Measure (LNC)

[Leacock, et al. 1998] measures only between noun concepts by following IS-A relations in the WordNet1.7 hierarchy. LNC computes the shortest number of intermediate nodes from one noun to reach the other noun concept. This is a measurement that human can think intuitively about the semantic distance between two nouns. Unfortunately, WordNet1.7 has a different root node. Therefore, no common ancestor between two keywords can happen. To avoid that, LNC measure introduces the hypothetical root node which can merge multiple-root tree into one-root tree.

This similarity measure is introduced in [Leacock, et al. 1998]. The similarity between two concepts is defined as:

$$Sim_{lch}(c_i, c_j) = \log\left(\frac{length(c_i,c_j)}{2 \times D}\right) \tag{3.13}$$

Where c_i, c_j are the concepts, $length(c_i, c_j)$ is the length of the shortest path between concepts c_i and c_i using node counting and D is the maximum depth of the taxonomy. Shortest Length means the shortest path between two concepts. D is the overall depth of WordNet1.7 and a constant value of 16.

3.3.3.5 Lesk Measure (LESK)

In Lesk measure [Lesk, et al. 1986] similarity of two concepts is defined as a function of overlap between the definitions of the concepts provided by a dictionary. It is described as:

$$Sim_{lesk}(c_i, c_j) = \frac{def(c_i) \cap def(c_i)}{def(c_i) \cup def(c_j)} \tag{3.14}$$

Where $def(c)$, represents the words in definition of concept c. This measure is not limited to semantic networks, it can be computed using any electronic dictionary that provides definitions of the concepts.

3.3.3.6 Wu & Palmer Measure (WUP)

This similarity metric [Wu, et al. 1994] measures the depth of two given concepts in the taxonomy, and the depth of the LCS of given concepts, and combines these figures into a similarity score:

$$Sim_{wup}(c_i, c_j) = \frac{2 \times depth\left(LCS(c_i, c_j)\right)}{depth(c_i) + depth(c_j)} \tag{3.15}$$

Where $depth(c)$ is the depth of the concept c in the taxonomy, and $LCS(c_i, c_j)$ is the LCS of the concepts c_i and c_j.

3.3.3.7 Hirst & St-Onge Measure (HSO)

This measure is a path based measure, and classifies relations in WordNet as having direction. For example, is-a relations are upwards, while has-part relations are horizontal. It establishes the similarity between two concepts by trying to find a path between them that is neither too long nor that changes direction too often. This similarity measure is represented with Sim_{hso}. Detailed description of this method can be found in [Hirst, et al. 1998].

Comparison of the Measures

Every measure has some shortcomings. On the one hand, RIK measure cannot differentiate the two keywords which have the same LCS. On the other hand, JNC and LIN address this problem. Their measures give the different similarity value of a pair of keywords having a same ancestor by considering its IC. However, JNC and LIN are sensitive to the Corpus. Based on Corpus, JNC and LIN may end up with different values. Furthermore, LNC measure has additional limitation. For some keywords, SL (Shortest Length) value does not reflect true similarity. For example, furniture will be more closely related with door as compared to sky. However, with LNC, SL for furniture and door and SL for furniture and sky will be 8 in both cases. Due to the structural property of WordNet, it is quite difficult to discriminate between such keywords with LNC. The LSK measure uses the dictionary approach, while WUP is based on the depth of the concept in the taxonomy. The last measure method HSO performs the semantic similarity on the basis of path relation in upward directions which makes them costly in term of computation.

Each of the above approaches has their own benefits and restriction. For our research, we have selected only four of the methods (RIK, JNC, LIN and LNC), where first semantic similarity between the concepts are calculated individually and then their mean average are calculated to take maximum benefit from all of the four. For example, the semantic similarities between four randomly selected words (Sky, Water, Tree, Flower) by using the JNC measure, the semantic similarity of these concepts are presented in the Table 3.1, where we can easily judge that the semantic relevancy among the terms. The semantic similarity values among the terms fluctuate between 0 and 1, the value approaches to 1 delineates the greater relevancy, while the value approaches to 0 represents the fewer relevancies. In the below Table 3.1, the semantic similarity between the same terms is 1 like sky which is the maximum semantic similarity value, while the semantic similarity between Tree and Sky is 0.1625 and between Tree and Water is 0.2232, while among Tree and Flower is 0.4742. Among the terms the Tree is most related with the Flower instead of Sky and Water and this can be represented by the semantic similarity values as well.

Table 3.1: Semantic Measure between the concepts using JNC Measure

	Sky	**Water**	**Tree**	**Flower**
Sky	1	0.1952	0.1625	0.1512
Water	0.1952	1	0.2232	0.2024
Tree	0.1625	0.2232	1	0.4742
Flower	0.1512	0.2024	0.4742	1

We have developed a *WordNet.SemSim()* function, that take the two words as an input and return the result as a semantic similarity. The following is the algorithm for semantic similarity calculation for each of the term in the list, the input and output are in the form of structure.

Propose Algorithm 3.3: Calculating Semantic Similarity

Input: *L→ Concept, Synset.name, ConceptSet.name*

Output: *L → Concept, Synset.name, SynSet.SS, ConceptSet.name, ConceptSet.SS*

Method:

i → Length (L)

//calculating and adding Semantic Similarity for Synset

j → Length(L(i).Synset)

L(i).Synset(j).SS ← WordNet.SemSim(L(i).name, L(i).Synset(j).name)

//calculating and adding Semantic Similarity for Synset

k → Length(L(i).ConceptSet)

L(i).ConceptSet(k).SS ← WordNet.SemSim(L(i).name, L(i).ConceptSet(k).name)

3.3.4 Concept Refinement

The expanded form of the annotated document in lexical and commonsensical dimension comes up with too many keywords; some of them are relevant and some are irrelevant which decrease the precision of the query. In order to achieve the precision, we have to remove these noisy keywords. One of the main challenge in this regard is to decide that which one of the keywords has to be removed and which one has to be included. In order to put the appropriate words or concepts in the annotation documents, we consider the semantic similarity values as calculated and store in the previous module, we defined a threshold value for the candidate term selection, which is in this case is 0.60. The sematic similarity values among the original and any of the expanded term above this threshold are eligible for a candidate terms selection while rest of the keywords are discarded. By doing this, we achieve significantly increase in precision even for the worst queries. After the annotation refinement and validation, the equation (4) becomes

$$C' = \bigcup_{j=1}^{m} \left(\bigcup_{i=1}^{n} (t', aE') \right)_j \tag{3.16}$$

The algorithm for candidate terms selection is as under,

Propose Algorithm 3.4: Candidate Concept Selection

Input: *L → Concept, SynSet.name, SynSet.SS, ConceptSet.name, ConceptSet.SS*

Output: L_{fp} *→ Concept, Synset.name, SynSet.SS, ConceptSet.name, ConceptSet.SS*

Method:

th ← 0.60
i → Length (L)

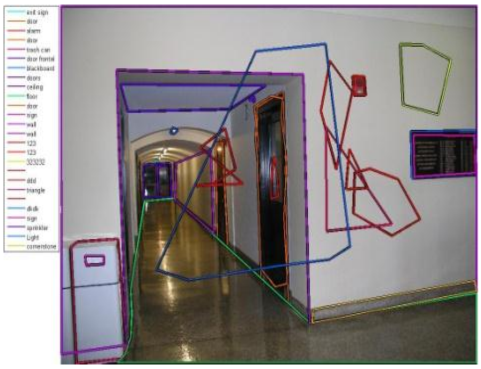
$L_{fp}(i).name \leftarrow L(i).name$
$ind \leftarrow 0$

//Candidate terms selection from SynSet
j→Length(L(i).Synset)
$IF\big(L(i).SynSet(j).SS \ \geq th\big)THEN$
$L_{fp}(i).SynSet(++ind) \leftarrow L(i).SynSet(j)$

//Candidate terms selection from ConceptSet
$ind \leftarrow 0$
k→Length(L(i).ConcepSet)
$IF\big(L(i).ConceptSet(k).SS \ \geq th\big)THEN$
$L_{fp}(i).ConceptSet(++ind) \leftarrow L(i).ConceptSet(k)$

Table 3.2 shows the annotation results of two exemplary randomly selected images. The results show that the proposed framework performs well.

Table 3.2: Result of the Proposed Framework for the sample two images.

Image with original annotation		

<table>
<tr><td>Original Keywords</td><td colspan="2">pedestrian, ceiling, plant, light, light, trash can, doorway, sign, bench, bench, person, door, personStanding, personStanding, groupOfPeople, person dark, person, person, person occluded, person, person lowres, window, person walking, person dark, lamp, lamp, person sitting, chandelier, chandelier, Person sitting, window, sign, ceiling, column, corridor, floor, wall, window, wall, wall, 111, tree, column, column, bin, Penis, Jim, pot, column, sign, person occluded, Ketna, ccccccccccc, aszxaszx, floor, chain, chain, bulb, text, column</td><td colspan="2">exit sign ,door, alarm, door, trash can, door frontal, blackboard, doors, ceiling, floor, door, sign, wall, wall, 123, 123, 323232, ddd, triangle, dkdk, sign, sprinkler, Light, cornerstone</td></tr>
<tr><td>Image with annotation after DFP</td><td colspan="2">
</td><td colspan="2">
</td></tr>
<tr><td>After DFP Keywords</td><td colspan="2">bench[2], bin[1], bulb[1], ceiling[2], chain[2], chandelier[2], column[5], corridor[1], floor[2], person[14], pot[1], sign[3], text[1], tree[1], wall[3], window[3]</td><td colspan="2">alarm [1], ceiling [1], cornerstone [1], door [5], floor [1], light [1], sign [2], sprinkler [1], trash can [1], wall [2],</td></tr>
<tr><td rowspan="2">Synset & Conceptset Added</td><td>Synset</td><td>Conceptset</td><td>Synset</td><td>Conceptset</td></tr>
<tr><td>117</td><td>313</td><td>128</td><td>185</td></tr>
<tr><td>Refined Candidate Keywords</td><td colspan="2">bench, bin, bulb, cap, ceiling, chain, chandelier, column, corner, corridor ,floor, flooring, individual ,light bulb, mark, pendant, person, pot, rampart, sign, text, textual matter, toilet, tower, tree, wall, window, windowpane, augury, base, batch, bed, bench, bulwark, commode, container, corporation, crapper, deal, dope, editorial, flock, flowerpot, foretoken, gage, good deal, grass, hatful, heap, house, jackpot,</td><td colspan="2">basis, cap, ceiling, cornerstone, door, doors, doorway, floor, flooring, light, lightsome, mark, rampart, sign, sprinkler, trash, trash can, wall, alarm, alarm clock, alarm system, alarum, alert, augury, base, bed, bulwark, clock, consternation, dismay, foretoken, foundation, fundament, groundwork, house, level, lightheaded, mansion, match, parry, polarity, room access, sign of the zodiac, signal, signboard, star sign,</td></tr>
</table>

	judiciary, kitty, level, locoweed, long chair, lot, mansion, mass, medulla, mess, mint, mortal, mountain chain, mountain range, pane, passage, peck, pendent, pile, pillar, plenty, polarity, potato, potbelly, potentiometer, potty, raft, range, range of mountains, schoolbook, shoetree, sight, signal, signboard, skunk, slew, smoke, soul, spate, stack, star sign, stool, storey, story, strand, string, terrace, text edition, textbook, throne, tummy, wad, weed, whole lot, whole slew, workbench	storey, story, threshold, warning device, warning signal

The Table 3.2 shows the result of the proposed framework on the randomly selected two sample images from the LabelMe dataset. The images in the LabelMe are tagged with the list of objects which are represented by the set of polygons. The sample images consist of colored lines which represent the objects. In the sample image annotation some of the terms like cccccccc, aszxaszx and 111 are the noises, these doesn't contribute to the actual meaning or semantics behind the concepts. These noises are removed in order to select only those terms that reflects to the semantic idea behind the image. The filtration process will decrease the computational overhead for further expansion. The refined original concepts will then be expanded to capture all the possible interpretation of the image semantics. The term expansion increases the recall of the system significantly but decreases the precision of system. In our proposed approach the SynSet and the ConceptSet expands the number of concepts tag with the image. But among the expanded terms, all the terms doesn't contribute a lot. These expanded terms are prune from the noises or less relevant terms by using the semantic similarity function. This semantic similarity computation will maintain the precision of the system. Among expanded terms, the candidate terms are made setting a threshold between the original terms after the filtration. The thresholds are computed by taking the average mean between the refined tagged concepts and the expanded concepts.

3.4 Experimental Setup and Evaluation

All experiments and evaluation of proposed framework have been performed on the LabelMe datasets, available freely for research created by the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) which provides a dataset of digital images with annotations. As of October 31, 2010, LabelMe has 187,240 images, 62,197 annotated images, and 658,992 labeled objects. The LabelMe dataset is dynamic, free to use, and open to public contribution. LabelMe was originated to figure out several common inadequacies of available data. LabelMe data set comprises a large number of annotated images, with many objects labeled per image as shown in the Figure 3.7. The objects are often carefully outlined using polygons instead of bounding boxes. Table 3.3, shows the comparison of LabelMe datasets with other benchmark datasets for testing and evaluation of algorithms. However, for testing and evaluation, if other datasets are to be considered then their annotation file should be transform to LabelMe XML file format.

Table 3.3: Summary of datasets used for object detection and recognition research and suitable for this research work.

Dataset	Images	Annotation	Annotation Type
LabelMe	187,240	62,197	Polygons
Caltech-101 [Fei-Fei, et al 2007]	8765	8765	Polygons
MSRC [Winn, et al 2005]	591	1751	Region Masks
CBCL-Streetscenes [Bileschi, et al 2006]	3547	27666	Polygons
Pascal2006 [Everingham, et al 2006]	5304	5455	Bounding Boxes

The following are some of the characteristics of the LabelMe datasets that distinguish LabelMe from other datasets and suitable for research in this kind of work.

i. *Complex Annotation:* Despite labelling an entire image (which also limits each image to containing a single object), LabelMe allows annotation of multiple objects within an image by specifying a polygon bounding box that contains the object. The Figure 3.8 shows the number of objects per image.

ii. *High quality labeling:* Countless databases just provide captions, which stipulate that the object is existing somewhere in the image. However, as argued about, more detailed information, such as bounding boxes, polygons or segmentation masks, is tremendously helpful.

iii. Contains a sizeable amount of object classes and permits the creation of new classes easily.

iv. *Diverse images:* LabelMe contains images from many different scenes, which demands for the non-domain specific approach.

v. Provides non-copyrighted images and allows public additions to the annotations, which provide an opportunity to do work on the real problem.

We investigate the performance of our system on three grounds, namely (1) how well it annotate the image semantically (i.e. Concept Diversity) (2) how well it prune the noisy tags from the annotation and how much increase occur in the re-annotation (i.e. Enhancement Ratio) (3) how much improvement it achieves in terms of image search and retrieval (i.e. Retrieval Degree).

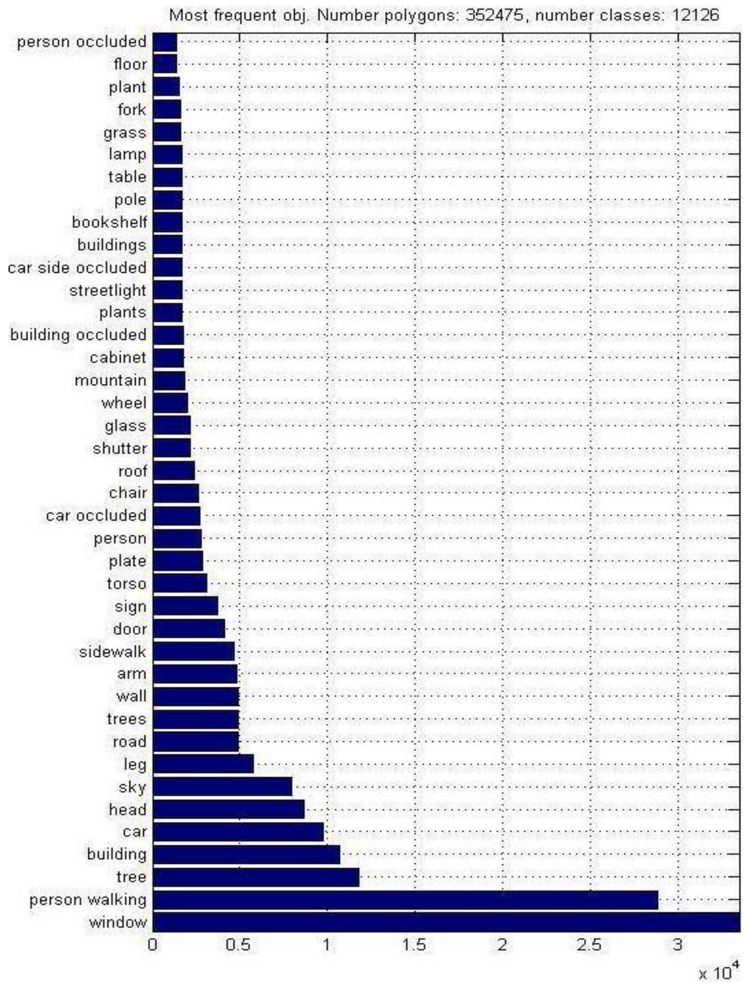


Figure 3.7: Shows frequency of objects in the LabelMe Datasets. The result is based on the datasets upto july 23, 2010.

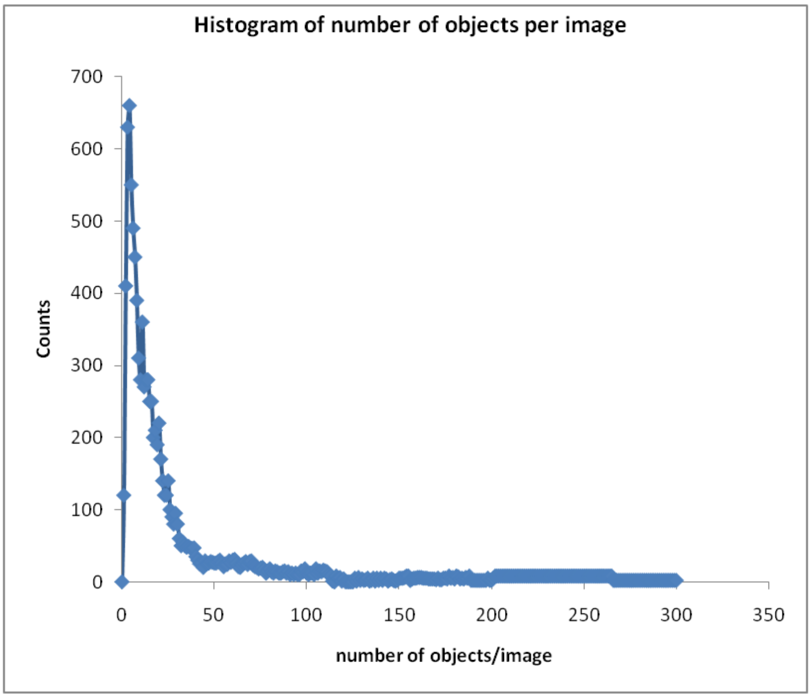


Figure 3.8: Show histogram of number of objects per image in the LabelMe Database

3.4.1 Concept Diversity

The concept diversity of annotations expresses the different topics or concepts exist in the dataset. In the LabelMe dataset, most of the user provides tags or keywords for the objects at the basic level of semantics, for example, the object like *'car'* is annotated as *'car'*, and while the upper level of semantics like *'vehicle', 'automobile', 'transport'* are ignore. We achieve a good improvement in concept diversity by adding the upper level of semantics along with other concepts from the commonsensical knowledgebases.

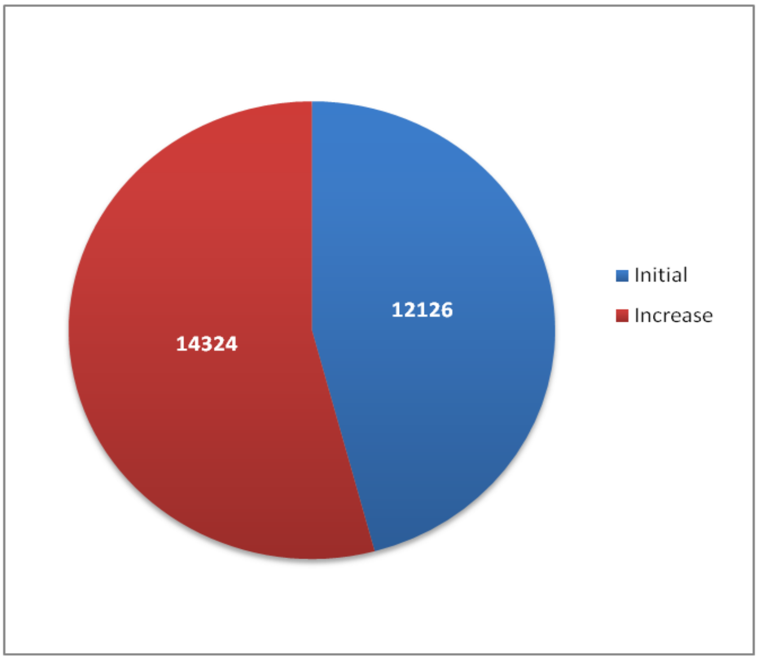


Figure 3.9: shows the Concept Diversity achieved after annotation enhancement and refinement perform over the LabelMe datasets

Figure 3.9 demonstrate the enhanced concept diversity of all differentiated tags. The enhanced concept diversity captured all the possible semantic interpretation of the image. The greater the concept diversity, greater is the semantic space for the images. The enhancement in terms of concept diversity is achieved through the proposed framework, where every single term (concept) already tagged with the image are expanded lexically and commonsensically through the phase of annotation enhancement using knowledgebases (see section 3.3.2). The initial terms tagged with the images included the noisy terms as well, which is further purified through the data filtration process (see section 3.3.1). The improvement in terms of concept diversity clearly depicts that semantic space of the images increase after the lexically and commonsensical term integration. This increase in the concepts classes are due to the expansion along with the refinement phase. It has been raised in a noticeable degree, i.e. from 12126 numbers of classes to 14324 and achieves 18.13% increase in the topic indexed.

3.4.2 Enrichment Ratio

Tagging ratio, which is the average number of labels tag per image, and enhancement ratio, which is the ratio of tagging ratio increase after enhancing and refinement annotation, formulas are explained in following equations,

$$T_1 = \frac{\sum_{i=1}^{n}(C_i)}{N} \tag{3.17}$$

$$T_2 = \frac{\sum_{i=1}^{n}(C_i)}{N} \tag{3.18}$$

$$T_3 = \frac{\sum_{i=1}^{n}(C_i)}{N} \tag{3.19}$$

Where T_1 is the tagging ratio before data filtration process (see section 3.3.1), T_2 is the tagging ratio after data filtration process (see section 3.3.1), while T_3 is the tagging ratio after enhancement & refinement while C_i is the number of concepts tag with the image respectively.

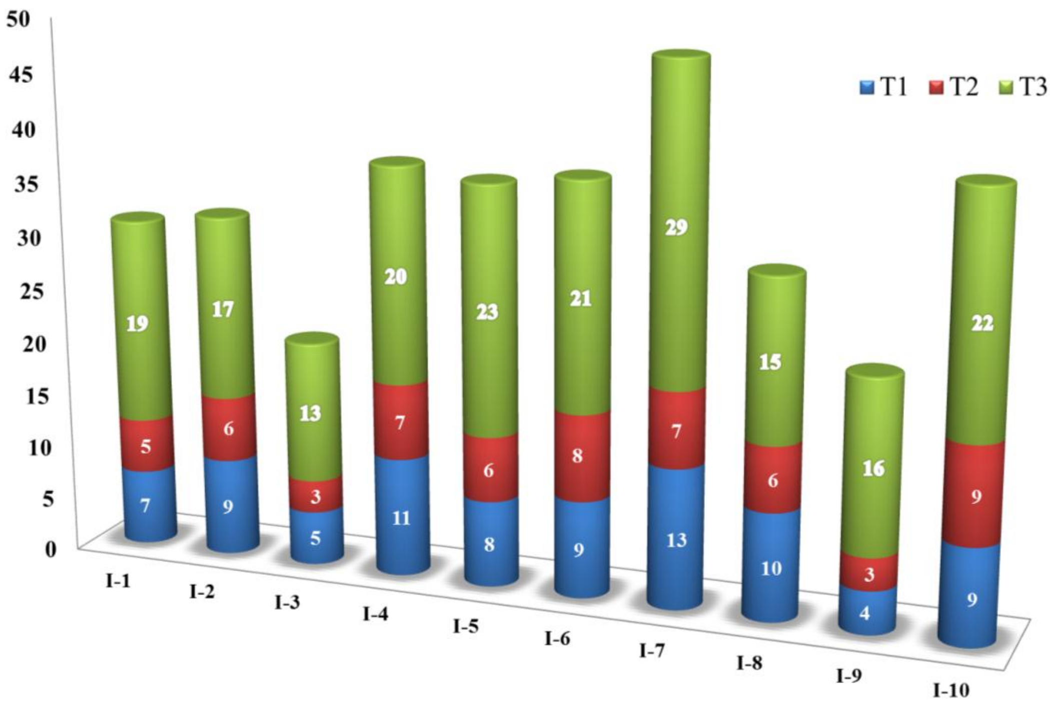


Figure 3.10: Graph shows the number of tags per image of the 10 sample images taken from the LabelMe dataset, where T_1 and T_2 represents the number tags before and after data filtration process, while T_3 shows number of tags after the annotation enhancement and refinement phase.

The Figure 3.10 depicts the tagging ratio of the randomly selected 10 sample images. Originally, the images were tagged with the terms, where some of the terms were unusual and noisy which is delineated by T_1In the proposed framework, the initial tag terms were first needed to be prune from these noisy terms (see section 3.3.1) and then the selected terms are passed to the next phase of the proposed framework i.e. the expansion phase (see section

3.3.2). The output of the initial refinement is represented by T_2. The refine tagged terms are then passed to the expansion phase to cover all the possible semantics dimensions of the images. The outcome increased in the tags per image of the expansion phase is delineated by T_3, which is the ratio between the refine and expanded lexical and conceptual terms. For instance, the image I_1 in Figure 3.10 is initially tagged with $T_1 \leftarrow 7$, these tags are then refined to $T_2 \leftarrow 5$. This decreases the number of tags as there were two unusual terms removed in the filtration process and filter out only those terms which contribute to the actual meaning behind the group of an object that constitutes an image. After the expansion, the number of tags per image became $T_3 \leftarrow 19$, which raised the tagging ratio 280%. Similarly the increase in the tag for $I_2 \leftarrow 183.33\%, I_3 \leftarrow 333.33\%,\ I_4 \leftarrow 185.71\%,\ I_5 \leftarrow 283.33\%,\ I_6 \leftarrow 162.5\%,\ I_7 \leftarrow 314.29\%,\ I_8 \leftarrow 150\%,\ I_9 \leftarrow 433.33\%\ and\ I_{10} \leftarrow 144.44\%$ respectively. The rate of an increase in the tagging ratio for the 10 sample images is different. It is because some of the images are simple while some of them are semantically enriched. The concepts in the simple images are limited so their semantic space will be small and therefore, their expansion will be limited. While for the semantically enriched images consist of a large number of concepts and constitute a large semantic space as a result, the percentage increase in the tagging ratio will be large, because, the expansion is applied on every single term of the filter out terms lexically and commonsensically.

As tagging ratio for the overall has risen from 6.19 tags per image in the dataset to 13.54 tags after annotation enhancement and refinement, whilst an enrichment ratio has achieved a considerable degree about 118.74%. There is although 2.90 unusual tags per images were removed or corrected by unification module.

The enrichment ratio is the ratio between the tagging as expressed in the equations below.

$$E_1 = \frac{T_2}{T_1} \tag{3.20}$$

$$E_2 = \frac{T_3}{T_2} \tag{3.21}$$

Where E_1 is the enrichment ratio for the T_1 and T_2, while E_2 is the enrichment ration for T_2 and T_3 respectively.

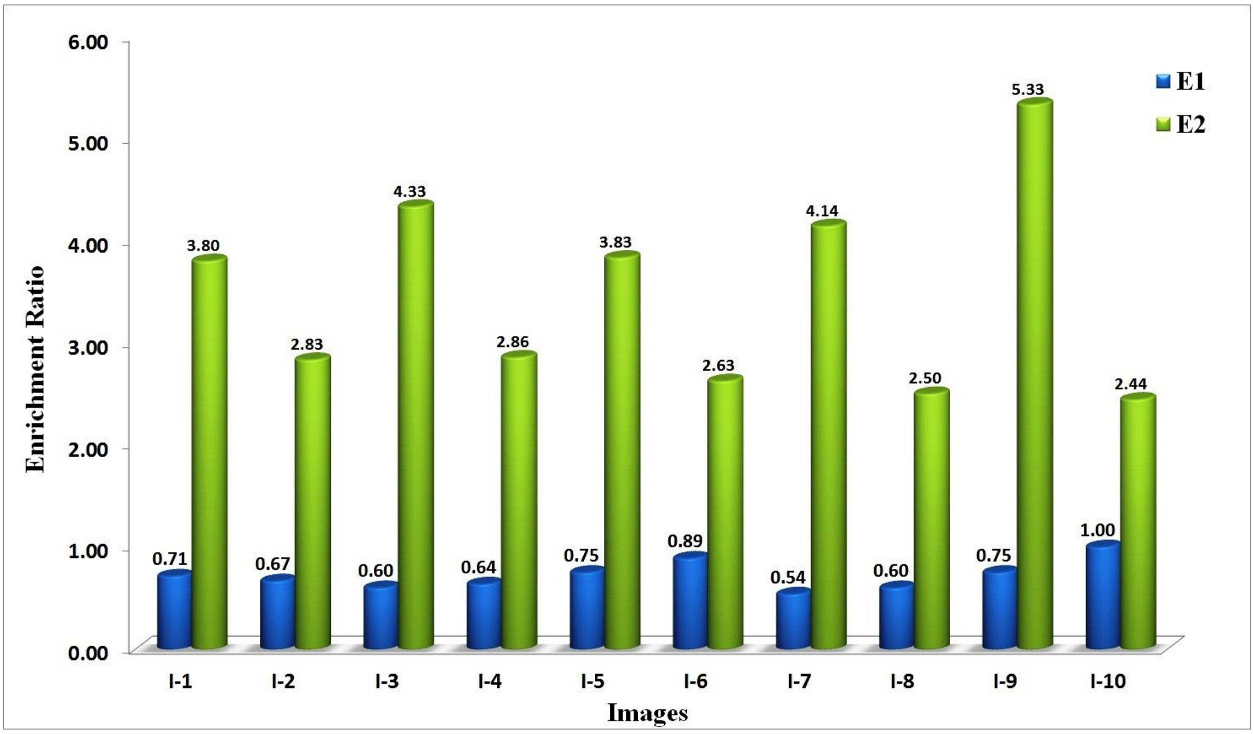


Figure 3.11: Graph shows the Enrichment ratio between the $\boldsymbol{E_1}$ *and* $\boldsymbol{E_2}$ *before/after the processing of the proposed framework*

The Figure 3.11 shows the enrichment ratio for the same randomly selected 10 sample images. The large gap among $\boldsymbol{E_1}$ and $\boldsymbol{E_2}$ are due to the fact, as the tags $\boldsymbol{T_1}$ are the baseline tags with the images, while $\boldsymbol{T_2}$ is the filter out representation of the same tags which is for the most images is same or less, so the enrichment ratio for this will always be either equal or less than 1. While for $\boldsymbol{E_2}$, the ratio is based on the $\boldsymbol{T_2}$ and $\boldsymbol{T_3}$, where $\boldsymbol{T_3}$ is representing the expanded tags which is for most of the images is greater than $\boldsymbol{T_2}$. So the enrichment ratio $\boldsymbol{E_2}$ will always be greater than or equal to 1. In the Figure 3.11, for example $\boldsymbol{I_9}$ have the highest $\boldsymbol{E_2}$ value among the others, which is due to the fact that the terms tag with the image $\boldsymbol{I_9}$ has a large number of lexical and conceptual expansion while the $\boldsymbol{I_{10}}$ have smaller $\boldsymbol{E_2}$ value is not only due to the small number of lexical and conceptual expansion but also the expanded terms are repeated, which were removed in the concept refinement phase (see section 3.3.4).

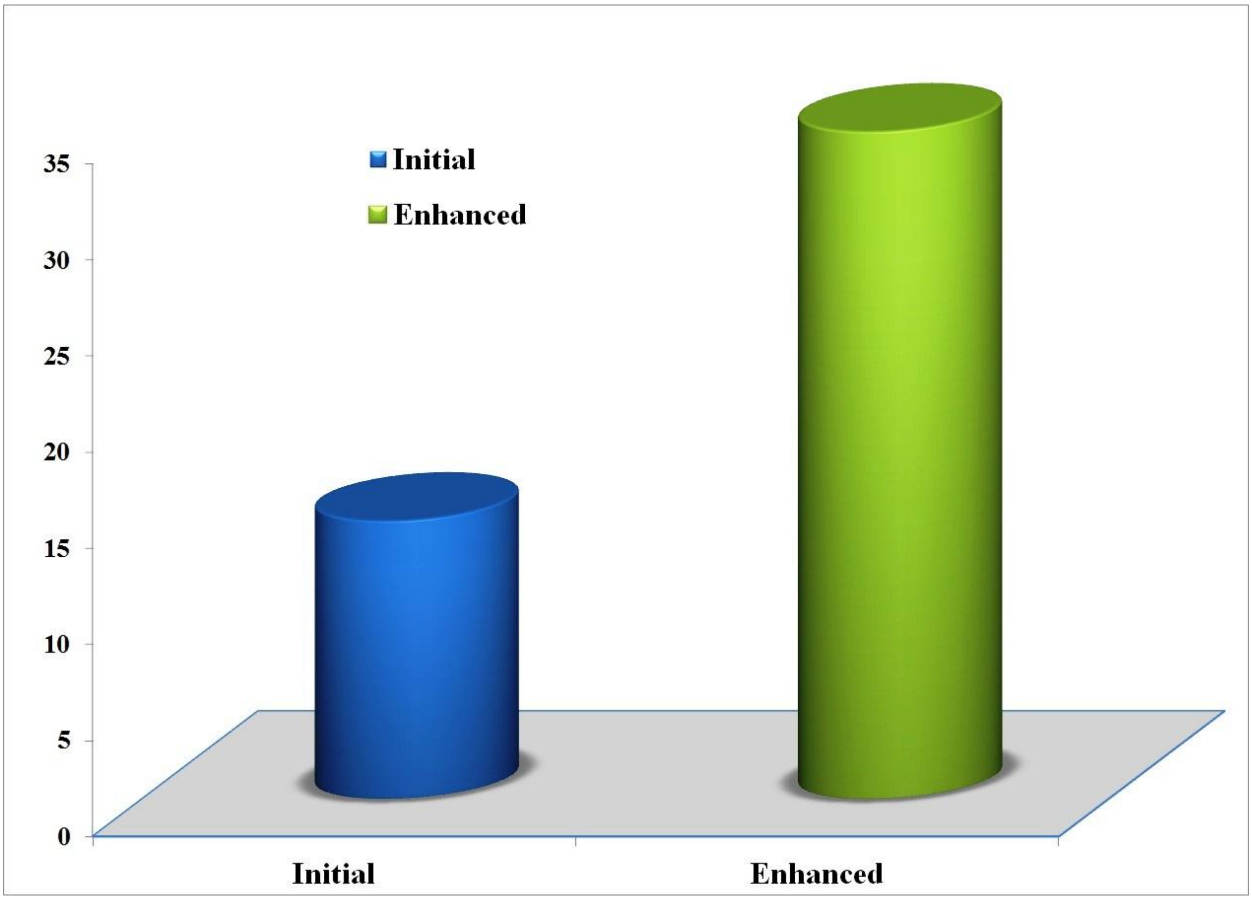


Figure 3.12: Graph depicts the overall Enrichment Ratio of the initial tags and tags after the enhancement. A considerable enhancement occurs in term of enrichability.

The Figure 3.12 shows the overall enrichment ratio before and after an annotation enhancement and refinement process. The initial graph represents the enrichment before the processing of the proposed framework, while the enhanced graph represents the enrichment achieved after the performing processing on the images datasets by using the proposed framework. It has been noticed during the process of the proposed framework that most of the terms have been repeated and needs to be controlled and pruned, which was further purified through concept refinement phase (see section 3.3.4). The concept refinement phase effectively controlled all the noisy terms generated through the expansion phase. The enhanced graph in the Figure 3.12 shows the purified form of the enrichment ratio achieved by the proposed framework. The Enrichment ratio achieves at higher level, because of the lexically and commonsensically expansion. The result of the Figure 3.10, 3.11 and 3.12 depicts the improvement in terms of an enrichment ratio. The higher the enrichment ratio, the higher the semantic space for the images and as a result increases the precision of the query even for a worst query as well.

3.4.3 Retrieval Degree

Retrieval degree is the number of correct images retrieved with a simple concept based query. We perform the experiments by using the LabelMe query engine, which work on the basis of string matching techniques for images search and retrieval in the LabelMe corpus. We use the retrieval degree of the LabelMe query engine as a baseline for the comparison. In Figure 3.13, the retrieval degrees of a different concept based queries are shown, the concept based query before and after enhancement & refinement. Using the proposed framework, the retrieval degree has been increased.

The Figure 3.13 depicts the retrieval degree of the randomly selected concepts from the LabelMe corpus. The selected concepts are either single concept words or multi-concept words. For instance, like *'car'* is a single concept word, while the concept like street is a combination of several other concepts like *road, tree, car, building* etc. The Figure 3.13 shows a significant improvement of the proposed technique over the baseline in terms of retrieval degree. It is due to the fact, that base line approach consists of a limited number of tags attached with the images. While the proposed approach attempts to cover all the possible dimensions of the semantic interpretation of the images, for instance, the first concept in the Figure 3.13 is *building*, which is a simple single concept word. The baseline approach only retrieve those images that are tagged with the keyword building regardless of other images that contain the same concept but are tagged with different word like *apartment, shopping mall, house* etc. even though both the concepts have same semantic meaning but different words. While our proposed technique attempts to tag all such types of words and concepts by using the lexical and commonsensical expansion (see section 3.3.2) in order to retrieve all the relevant images available in the corpus. All these expansion leads to the substantial improvement in term of retrieval degree.

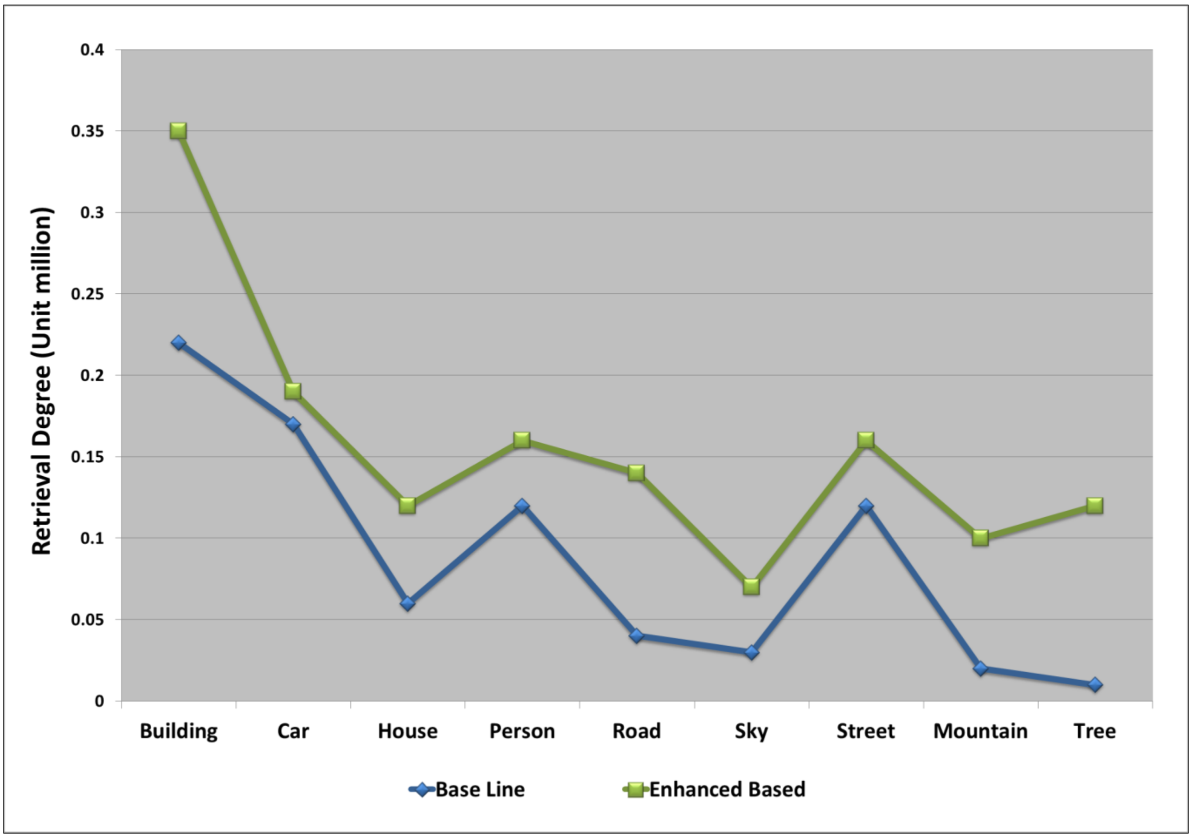


Figure 3.13: Graph shows the retrieval degree for the original and enhanced annotation performs on the LabelMe datasets. The results are produce by using the Query Engine of the LabelMe.

These results exhibits that searching and retrieval for images over enhanced annotation outperforms searching and retrieval using the original tags. In addition to that, annotation enhancement by the proposed framework surpasses the baseline approach in terms of concept diversity, enrichability, and most importantly retrieval performance.

3.5 Chapter Summary

In this chapter, we proposed framework for the image annotation enhancement and refinement framework. This framework makes use of lexical and commonsensical knowledgebases to enhance existing annotation for indexing in a superior way. Initially, the corpus is prune from redundant and noisy keywords while the expansions of the keywords are conducted by using the synset and conceptset via well-known knowledgebases WordNet and ConceptNet. The expanded form of the keywords come up with large number of unusual words, that need to be discarded, so the pruning of the expanded form of the data are done by the help of semantic similarity between original and expanded keywords. For evaluation, we perform all the experiments on LabelMe datasets for images. Results show that searching for an image over enhanced annotation outperforms using the original annotation. We achieve good results in terms of concept diversity, annotation enrichability and prominently retrieval performance.

This work is further extends to high level semantic propagation discussed in chapter 04, where the enhanced annotation are utilized to calculate the semantic similarity among the images on the basis of annotation.

Chapter 04

A Framework for High Level Semantic Annotation using Trusted Object Annotated Dataset

A photograph is a secret about a secret. The more it tells you the less you know.
Diane Arbus, photographer, 1923 – 1971

The ubiquitous multimedia data calls for efficient and flexible methodologies to annotate, systematize, warehouse and access multimedia resources. Multimedia annotation data plays an important role in the future annotation-driven multimedia system. Although the importance of the high level semantic (HLS) multimedia annotation data is widely recognized and a considerable amount of research has been conducted on its various aspects, there is no consistent framework on which to structure HLS multimedia annotation data. The HLS annotation of resources in general and multimedia resources in particular, is a resilient job. The progression in automatic annotation mechanisms have not been able to comprehend with adequately accurate results. To outfit multimedia (e.g. image) retrieval capabilities, digital libraries have hung on manual annotation of images. Providing a track to enact high level semantic annotation automatically would be more worthwhile, efficient and scalable with magnifying image collections. Since scarcity of storage space is not an issue, consumers have the opportunity of storing images without any consent to their quality and future use. Exploitation of these data requires an intelligent way to discover the desired image. The fast proliferation in the hard way technology also demands for the software for managing such an immense image collection. The main intriguing issue concerning the data mining and data management is retrieving the desired images. Researcher community is continuously striving for solving this dilemma.

The aims of this chapter is to take advantage from the previous work and propose a mechanism for the ease of manual annotation to a large pool of object annotated images datasets, where images are clustered based on the annotation and assigning high level semantic description to them. This sort of work can easily be applied on the LabelMe videos datasets and can be exercised on the web images and videos as well by integrating object recognition and specification components. This chapter intent to equip the high level semantic annotation for images, and consequently, contributes to 1) calculating semantic intensity (SI) of each object in the image depicting the dominancy factor, (2) image similarity on the bases of SI and metadata tag with the images, and (3) image categorization approach based on the image similarity to tag set of images with a high level semantic description with their calculated similarity values. The experiment on a portion of randomly selected images from LabelMe database manifests stimulating outcomes.

This chapter is organized as, in section 4.1 the introduction about the stated area is presented, while section 4.2 focuses on the state-of-the-art in the related area. Section 4.3 is dedicated to proposed framework, where a mechanism for semantic intensity (SI) which is concept dominancy factor in the image is discussed in detail, the algorithmic solution to each of the module is shown up. Adding to this, a brief overview on clustering is presented while semantic image similarity on the basis of annotation is discussing and is supported by example and at the end high level semantic propagation is discussed. The experimental work is discussed in section 4.4. The chapter is finally concluded with summary and future work in section 4.5.

4.1 Introduction

The latest trend in hardware and telecommunication technologies has resulted to a rapid growth of the available amount of multimedia information. Multimedia content is used in a wide range of applications in areas such as content production and distribution, telemedicine, digital libraries, distance learning, tourism, distributed CAD/CAM, GIS and of course on the World Wide Web. The usefulness of all these applications is largely determined by the accessibility of the content and as such, multimedia data sets present a great challenge in terms of storing, transmitting, querying, indexing and retrieval. To tackle such challenges it is not adequate for just developing faster hardware or to design more refined algorithms. Rather, a wiser understanding of the information at the semantic level is required [Chang, 2002]. This is of particular importance in many emerging applications such as semantic transcoding [Bertini, et al 2004], where it is assumed that the user does not want to access all data, but only data semantically useful. This requires the semantic identification of the objects and events appearing in the content so as to be in a position to match them with the user preferences. In this way, the part of the content which is of interest to the user is identified, isolated and transmitted.

In spite of the fact that new multimedia standards, such as MPEG-4 and MPEG-7 [Chang, et al 2001], provide the essential functionalities in order to manoeuvre and impart objects and metadata, their extraction, significantly at a semantic level, is out of the scope of this dissertation and is left to the content developer. In the last two decades, significant results have been reported regarding the successful implementation of several prototypes, [Salembier, et al 1999]. However, the lack of precise models and formats for object and system representation and the high complexity of multimedia processing algorithms make the development of fully automatic semantic multimedia analysis and management systems a challenging task [Chang, 2002]. This is due to the hardship, often concerned to as the semantic gap, of taking concepts mapped into a set of image and/or spatio-temporal features that can be automatically extracted from video data without human intervention [Al-Khatib, et al 1999]. Unfortunately, the result of the automatic annotation is far from satisfactory because of the large gap between low-level features and high-level semantics and the manual annotation is not only labor extensive and time-consuming for large multimedia data achieve, but also subject to human errors, the figure as shown in the Figure 4.1., where a comparison between human expert and machine annotation procedure, where human experts produce high level semantics of the multimedia directly, while the errors occurs in such a process are due to the human expert nature, as they apply similar approach for every multimedia document and violating the rich semantics inside the documents. On the other hand, the machine produces annotation at low-level and high-level, the error using this approaches are mostly occur due to the algorithm and techniques used, but these approaches are domain dependent, while human experts are domain independent.

Image retrieval has been extensively studied for many years and can be classified into text-based image retrieval (TBIR) and content-based image retrieval (CBIR). Content based image retrieval seems to be the most intuitive way of retrieving the images by employing the low-level features for extracting the semantics inside the image. CBIR is striving to reduce the semantic gap by relying on the low-level-features like color, shape and texture. Though these features can successfully interpret the contents of the image but flush out in interpreting the intended concept delineated by the image. While the TBIR relying on the metadata tag with the image. The TBIR strives to explore the semantics by applying the text mining techniques to the metadata. However, due to the ease of interpreting the user’s needs in natural language, the TBIR catches worthwhile researcher attention. All these considerations revealed that annotating the images with the appropriate concept and then classifying these images will be the predominant concern in multimedia management and retrieval. Manually annotating the images is a laborious task and quite seems impractical. Despite the fact, manual annotation seems to be the most instinctive way to describe the semantics of the image and this can be easily well performed by factoring and engineering the manual annotation process with the help of software. In the next section, we discuss the related work and state-of-the-art in this domain.

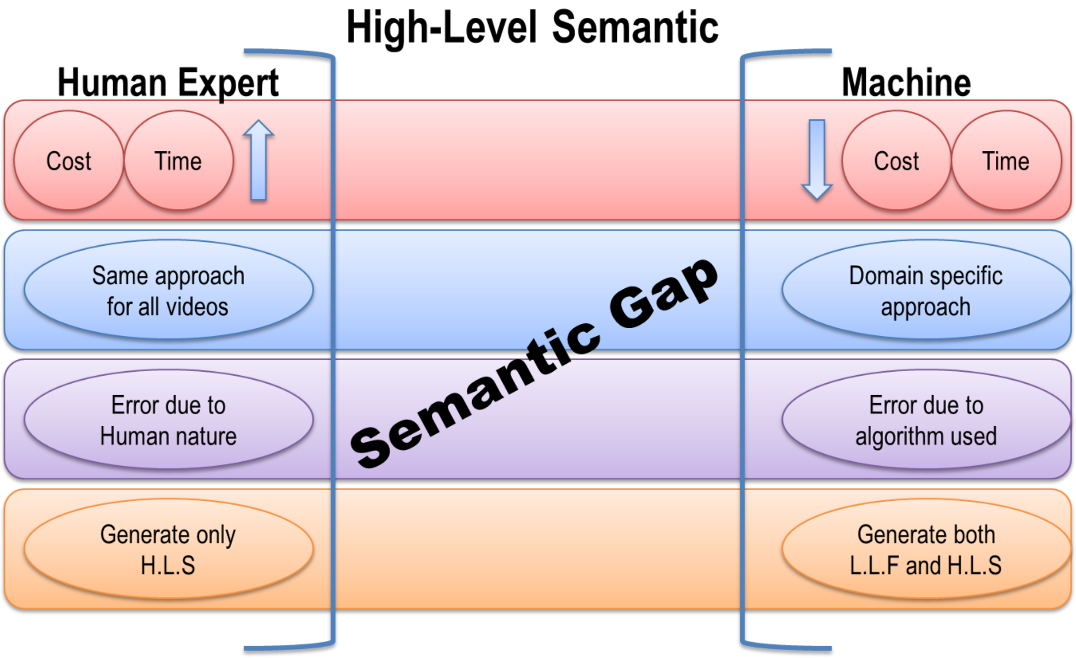


Figure 4.1: A typical comparisons of the human and machine annotation approach. (1) where human experts generates only high-level-semantics, while machine produce both low-level and high-level semantics,(2)the errors occur during the annotation process by human experts are due to their nature while machine produce due to the errors in the algorithm or techniques used. (3) Human experts used similar approach for all domain, while machine is domain dependent, (4) Human experts are costly and time consuming while machine is less time consuming and less costly.

4.2 State-of-the-Art

The growing number of digital images has brought about an urgent need to facilitate the retrieval and browsing of images via semantic keywords. Thus, techniques for Automatic Image Annotation (AIA) become increasingly important and a large number of machine learning techniques have been applied along with a great deal of research efforts. However, AIA task presents unique problems such as multi-label classification [Kang, et al 2006], and large scale concept space [Naphade, et al 2006]. These problems make AIA different and challenge for many traditional machine learning techniques and exacerbate the problem of semantic gap.

Many image annotation methods based on various learning techniques have been proposed in the literature. We can roughly classify these methods into two categories. The first category treats each semantic keyword or concept as an independent class and trains a corresponding classifier based on the training set to identify images belonging to this class, while the second category is based on the mapping of semantic keywords, where a correlation of keywords and their corresponding low-level features are calculated. The earlier efforts in this category were applied to extracting specific semantics, such as the work of the differentiating indoor from outdoor scenes by multi-stage classification approach where classifying the sub-blocks independently and then performing another classification on output of the previous result [Szummer, et al 1998], similar effort has been observed in [Vailaya, et al 1998], where classification between cities from landscapes are perform on the basis of low-level feature geared for the particular classes, they developed a procedure to qualitatively measure the saliency of a feature towards a classification problem based on the plot of the intra-class and inter-class distance distributions. By doing this, they determine the discriminative power of color-histogram, color-coherence vector, DCT coefficient, edge direction histogram and edge direction coherence vector. [Haering, et al 1997], works for detecting trees by combining colour measures and estimates of the complexity, structure, roughness and directionality of the image based on entropy measures, grey level co-occurrence matrices, Fourier transforms, multi-resolution Gabor filter sets, steerable filters and the fractal dimension. A neural network is then applied to arbitrate between the different measures and to find a set of robust and mutually consistent *"tree"*. [Forsyth, et al 1997] apply similar techniques for detecting the human and horses in the images by applying the statistical learning techniques to train the system to learn body plans in the images, their system demonstrates excellent performance on large, uncontrolled test sets. [Li, et al 2002] detects building in the images using the low-level features of extracted line segments and assigns them to consistent line clusters, new mid-level features that are used for high-level object detection and location, the proposed works well in classifying images of an independent web-derived dataset as building or non-building.

The representative technique for the first category is the classification technique such as the Support Vector Machine (SVM), neural networks, nearest-neighbor etc. which demonstrates strong discrimination power. The problem with the classification-based techniques is that they are not very scalable to large scale concept space. In the field of AIA, the semantic space is growing larger and larger along with more structural information. For instance, the widely used Corel data set contains more than 374 semantic labels [Duygulu, et al 2002], while the goal of LSCOM project is to build a semantic space consisting of thousands of concepts with rich semantic connections [Naphade, et al 2006]. Therefore, the problem of semantic overlap and data imbalance among different semantic classes induced by the multi-label characteristics of AIA is becoming more serious. Consequently, the classification power of this kind of approach is heavily impaired. Other methods in this category include [Yang, et al 2006] which performs image annotation with the help of multiple-instance-learning where the image is first segmented into regions and then apply the asymmetrical support vector machine false positives and false negatives, [Gao, et al 2006] achieve higher prediction accuracy for image classification and object class recognition by using a hierarchical boosting framework by incorporating the features hierarchy and boosting to scale up SVM image classifier training. [Amaral, et al 2010] works for hierarchical medical image annotation based on three different approaches using global and local features together with SVMs.

The second category of AIA methods focuses on learning the correlations between the visual features and semantic concepts. Many such methods are based on the generative model, in which an influential work is cross media relevance model (CMRM) [Jeon, et al 2003], which tries to estimate the joint probability of the image's visual keywords and the semantic keywords on the training set, but CMRM faces problem like it vector quantized the image regions into image blobs and this can reduce discriminative capability of the whole model. This problem was subsequently improved through a continuous relevance model (CRM) by preserving the continuous feature vector of each region and this offers more discriminative power [Lavrenko, et al. 2004], multiple Bernoulli relevance model (MBRM), which works on the existence/nonexistence binary status of each words, [Feng, et al 2004], the difference between the MBRM and CRM is the existence of a concept rather than its prominence. In differentiation to the conventional relevance models which calculate the joint probability of words and images over a training image database, the dual cross-media relevance model (DCMRM) model estimates the joint probability by calculating the expectation over words in a predefined lexicon [Liu, et al 2007]. The DCMRM involves two kinds of critical relations in image annotation. One is the word-to-image relation, and the other is the word-to-word relation. Both relations can be estimated by using search techniques on the web data as well as available training data. There are also efforts to consider the keyword correlations in the annotation process, such as the Coherent Language Model (CLM), that takes into account the word-to-word correlation by estimating a coherent language model for an image [Jin, et al 2004]. The problem with CLM is that they are unable to exploit correlations between class labels, for this the Correlated Label Propagation (CLP) proposed by [Kang, et al 2006], that explicitly models interactions between labels in an efficient manner by simultaneously co-propagates multiple labels. [Jin, et al 2005, Shi, et al 2006], put forward the translation model hybrid measuring (TMHD) model for improving the annotation using semantic similarity measure among annotated keywords and discarded the irrelevant words from the annotation by combining the evidence-rule based on the semantic similarity in WordNet. [Zhou, et al 2007] proposed the keyword correlations based concept annotation, where the correlation between keywords are analyzed by *"Automatic Local Analysis"* of text information retrieval. The Web sources are also exploited to improve image annotation [Liu, et al 2007]. Recently, [Qi et al. 2007] proposed a correlative multi-label (CML) annotation framework which simultaneously classifies concepts and models their correlations for video annotation.

The generative based (visual features & keywords) methods have shown better durability to the scalability of concept space, and provides a natural ranking for choosing the proper keywords as semantic annotations. However, many such methods are based on the strong assumption that visual similarity guarantees semantic similarity which is often violated as a consequence of the well-known semantic gap problem. For instance, images belonging to the same visual neighborhood often do not share similar semantic contents. In fact, the semantic gap problem implies that similar visual contents may correspond to multiple different semantic meanings. It is one of the reasons that the intuitive approach of designing a "good" metric measurement or density estimation method to directly bridge the semantic gap does not lead to satisfactory results.

The above discussions highlighted the key challenges improving the performance of AIA task. The challenges include (1) the ability to scale up the large concept space, and (2) the mismatch between visual similarity and semantic similarity. Hence the trend moves from automatic to semi-automatic approaches, where the researcher used the advantages of the both automatic and manual annotation techniques. [Wenyin et al. 2001], describes a semi-automatic image annotation process that is better than manual annotation in terms of efficiency and better than automatic annotation in terms of accuracy. The strategy aims to combines content-based image retrieval and user verification to achieve correct high-level metadata, i.e. to create and refine annotations by *"encouraging the user",* to give relevance feedback, [Lu et al. 2000] of the retrieved results. [Ivan et al 2010] described the object-based tag propagation technique for semi-automatic image annotation. [Yan SONG et al. 2005] proposed a semi–automatic video annotation strategy for video semantic classification, using relevance feedback to refine the classification, and active learning process to speed up the automatic learning process of classifying videos, by labeling the most informative samples. [M. Fischer, 2008], applied the semi-automatic techniques for face recognition for a TV series.

The flexible nature of semiautomatic annotation approaches makes it popular for small size of corpus or usually used for preparation of training data. Moreover, all of the above stated approaches use the keyword based annotation techniques and there is a limited work done for the semantic annotation or fixings the semantic in the annotation by semi / automatic means. To date, most of the research focuses on how to annotate the multimedia contents semantically, but still the high level semantics annotation is far from the satisfactory, because the way of multimedia understanding for human is not depend on keywords feature, but human would like to rely on their *"knowledge"* which came from previous personal experiences. To bridge the semantic gap, we should try to reflect the way of human perception for multimedia understanding. The manual annotation is the only source to date that can achieve this, but due to their laborious and costly nature is not feasible for large corpus. However, we can take advantages of the manual annotation and automatic annotation to form a semiautomatic environment, where the images are automatically categories with each other in an unsupervised manner by taking one image as a source and compute their semantic similarity with the rest of the images in the corpus and thus form a type of chain among the images in the corpus. On single effort for the high level semantic annotation for any given image, the other images in the chain automatically get the same annotation with their image similarity values. Moreover, the greater the similarities value the greater high level semantics for the images. In the next section, we have discussed the proposed framework for high level semantic propagation.

4.3 Proposed Framework

The proposed framework is based on the process of automatic classification and categorization of the images for the high level semantics propagation. In a big picture the proposed framework is divided into two parts. (1) Part-1: the LabelMe datasets are first purified and then semantic intensity of each object in the image are calculated, while (2) Part-II, the effort in this section is further divided into two sections (a) images similarity calculation among the selected with rest of the images in the corpus, the value for image similarity is fluctuated between 0 and 1. For every image we maintain two sets i.e. full similar (FS) and partial similar (PS), the images that have similarity value greater than 0.80 are put in the FS set, while images that have similarity value greater than 0.50 are part of the PS set. This process is repeated until the FS and PS sets for each of the images in the corpus are prepared. (b) High level semantic propagation is used to allow the human to annotate any image with the high level semantics in natural language which is more understandable to the human and system will automatically propagate this high level semantic to the rest of the images in the FS and PS sets. This process will continue until all the images in the corpus are annotated. The workflow of the proposed framework is presented in the Figure 4.2.

The LabelMe corpus is represented by using the equation 3.2 as,

$$C = \bigcup_{j=1}^{m} \left(\bigcup_{i=1}^{n} t_i \right)_j \tag{4.1}$$

Where *C* represent the union of all images in the corpus, while t_i is representing number of tags attached with each image in the corpus.

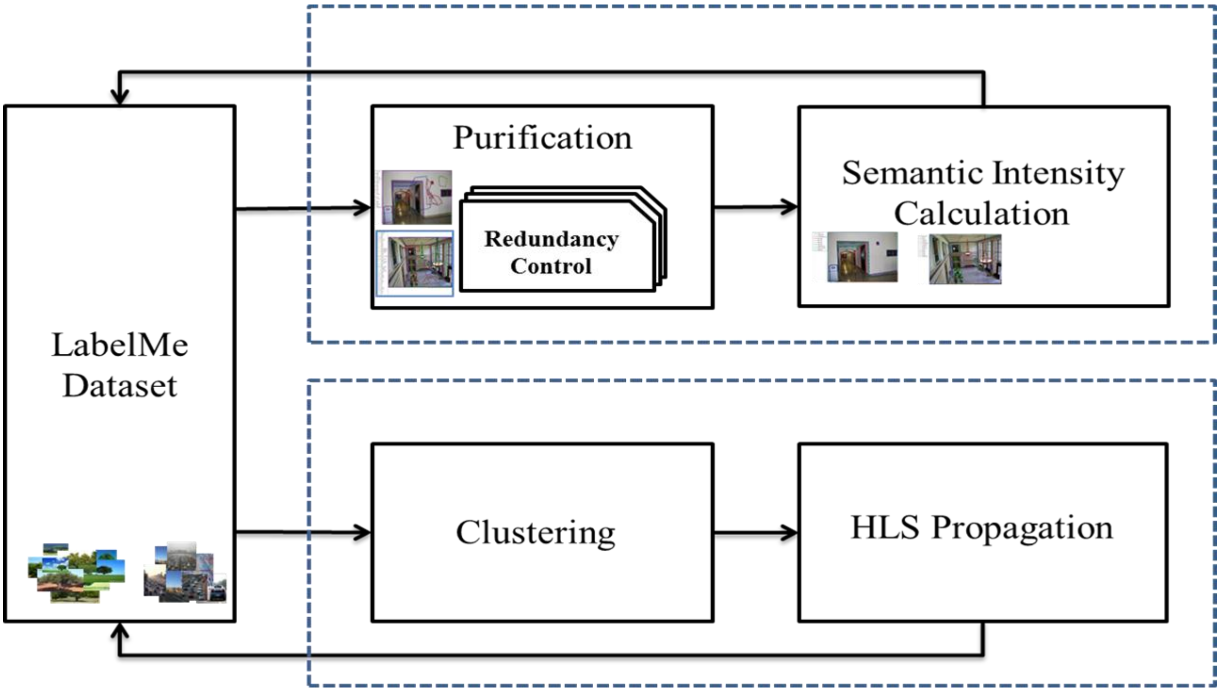


Figure 4.2: Proposed model for the high level semantic propagation

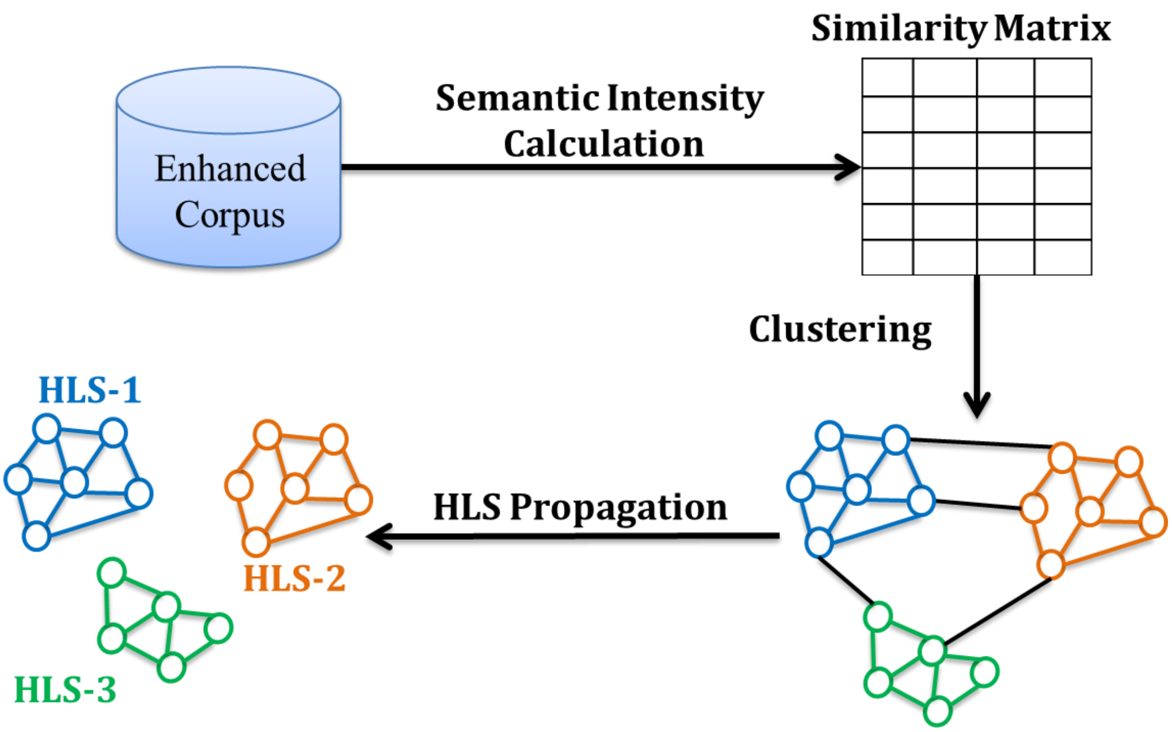
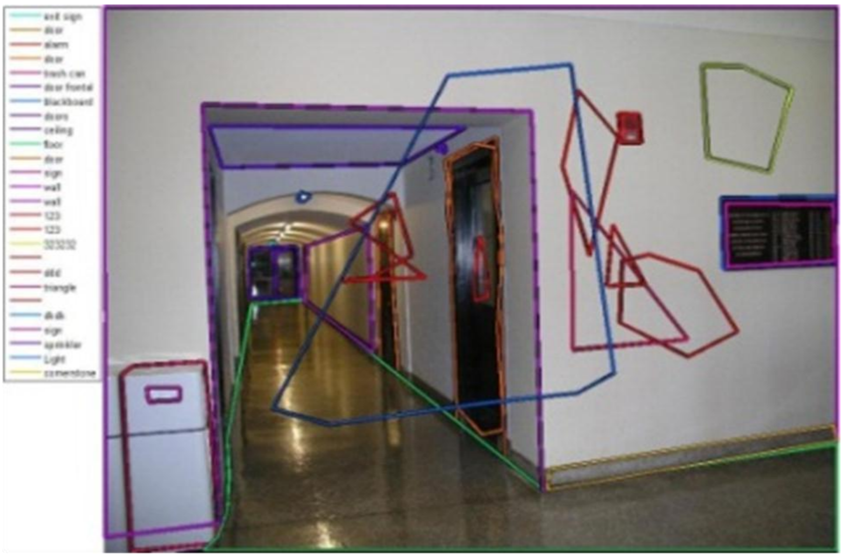


Figure 4.3: HLS propagation process, where semantic intensity of each concepts are calculated and then similarity matrix of the images are prepared, cluster are then prepared and then HLS description are assign to each of the images cluster.

4.3.1 Annotation Purification

The annotation of the LabelMe corpus is purified from irrelevant, unusual and redundant keywords by using the similar approach as discussed in the chapter 03, sections 3.3.1 under Data Filtration Process (DFP) head. The purified form of the corpus is represented by equation 3.5**,**

$$C' = \bigcup_{j=1}^{m} \left(\bigcup_{i=1}^{n} t_i' \right)_j \qquad (4.2)$$



(a) Image sample of the LabelMe before purification process

(b). Image with purified annotation, where number of instance for each object are represented in parenthesis.

Figure 4.4: Sample Image taken from LabelMe corpus before/after Annotation Purification

In the following sections, the proposed model is described in more details.

4.3.2 Semantic Intensity

"An image is worth of thousand words" [BOOK-1] clearly depicts the complex nature of the image and the dynamics of semantics inside the image. A single image depicts different semantic meanings based on the human perception. The Semantic Intensity can be defined as the *"concept dominancy factor with in the image"*. As images are the combination of different objects, these objects constitute to form different semantic idea. Different combination of objects depicts different concepts. However, these semantic ideas have different dominancy degree. Some of the ideas in the image are more dominant than the other as shown in the Figure 4.5.

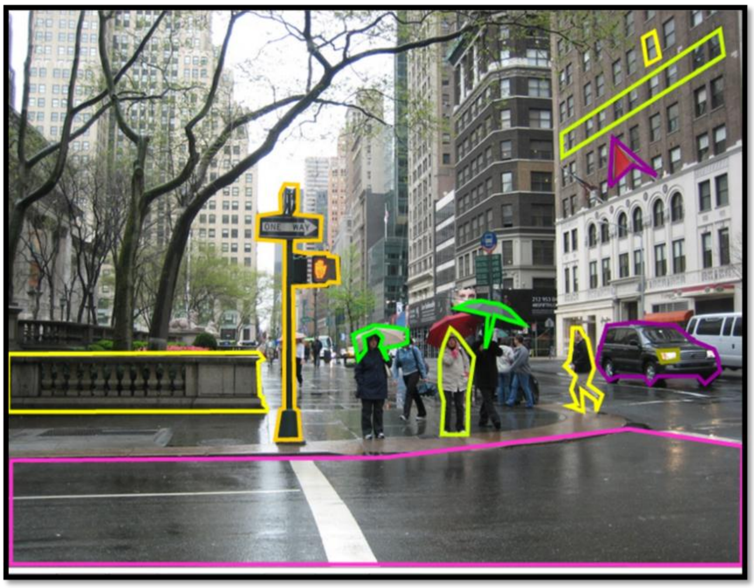


Figure 4.5: The image is taken from the LabelMe dataset. Image depicts a list of concepts like road, vehicles, signs, buildings, sky, trees, umbrella, buildings, street, cross walk, highlight, flags etc. and some hidden concept like rain. Among all the concepts some are more dominant like street, building etc.

a) Semantic Intensity Calculation

The web tool of LabelMe provide an opportunity to the users to annotate objects in the image by first sketch the border points and then tag with the user defines concept, object edges are represented in the form of polygon points in the annotated dataset as shown in Figure 4.5, the main drawback of the LabelMe web tool is that it provide a free hand to the user to sketch any object without considering the edges of the object and tag them with any concepts. This gives birth to a problem like irrelevant and unusual objects/concepts in the annotation. During the annotation purification process the remedies for these types of data are accomplish. Figure 4.6 shows the same image after purification, where unusual keywords and objects are filter out, while the XML representation of the annotation file of the LabelMe data are presented in the Figure 4.7, where each point of the polygon is represent. The SI value is calculated on the basis of these polygon points, but before calculating the SI value of the concepts in the image, a short discussion of polygon area calculation is presented.

The area A of a regular n-sided polygon having side s, apothem a, and circumradius r is given by

$$A = \frac{1}{2}nsa = \frac{1}{4}ns^2 \cot\frac{\pi}{n} = na^2 \tan\frac{\pi}{n} = \frac{1}{2}nr^2 \sin\frac{2\pi}{n} \tag{4.3}$$

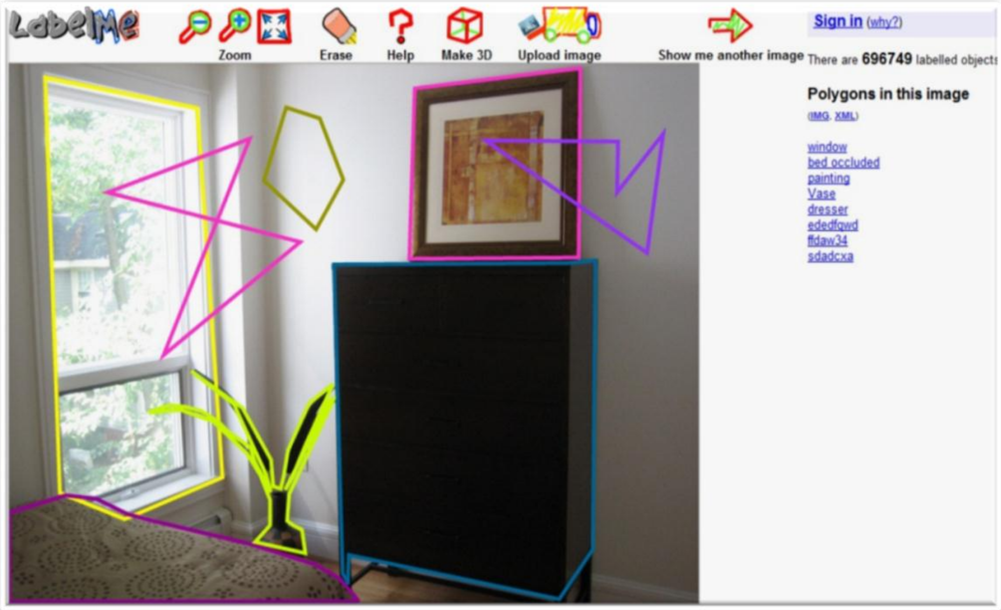


Figure 4.6: LabelMe web tool for images annotation, where some of objects are irrelevant and need to be discarded before processing

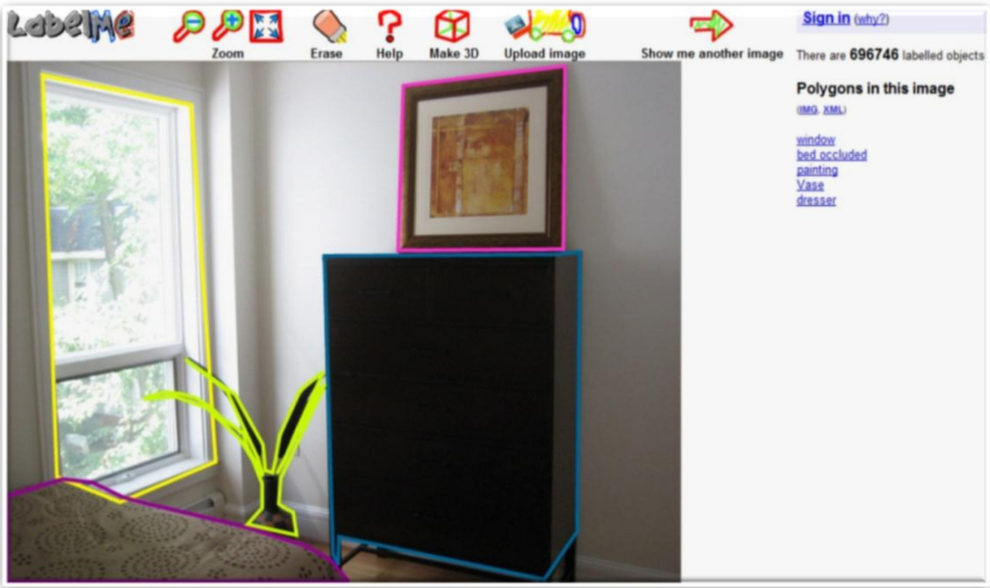


Figure 4.7: Show snapshot of the web tool of LabelMe, where each irrelevant and unusual objects and their tag words are removed.

```
<annotation>
        <filename>IMG_9785.jpg</filename>
        <folder>static_houses_boston_2005</folder>
        <source>
                <sourceImage>The MIT-CSAIL database of objects and scenes</sourceImage>
                <sourceAnnotation>LabelMe Webtool</sourceAnnotation>
        </source>
        <object>
                <name>window</name>
                <deleted>0</deleted>
                <verified>0</verified>
                <date>12-Feb-2009 14:06:39</date>
                <id>1</id>
                <polygon>
                        <username>anonymous</username>
                        <pt> <x>106</x> <y>40</y> </pt>
                        <pt> <x>558</x> <y>127</y> </pt>
                        <pt> <x>633</x> <y>1178</y> </pt>
                        <pt> <x>343</x> <y>1297</y> </pt>
                        <pt> <x>150</x> <y>1222</y> </pt>
                </polygon>
        </object>
```


Figure 4.8: Snapshot of the annotation file used by the LabelMe web tool for object edge representation

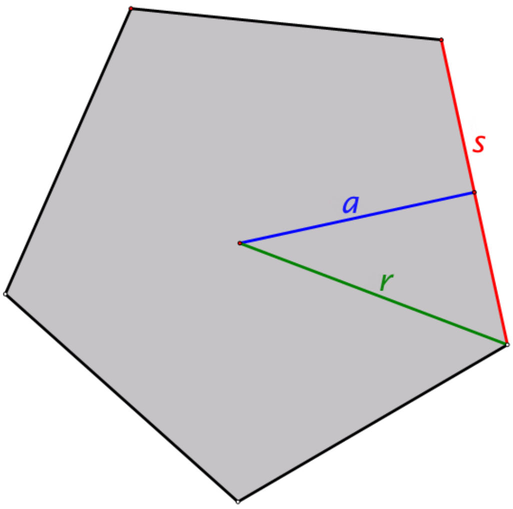


Figure 4.9: Shape of the regular [RP] polygon, with side s, apothem a and circumradius r.

While area of the irregular polygon is

$$A_{poly} = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) \qquad (4.4)$$

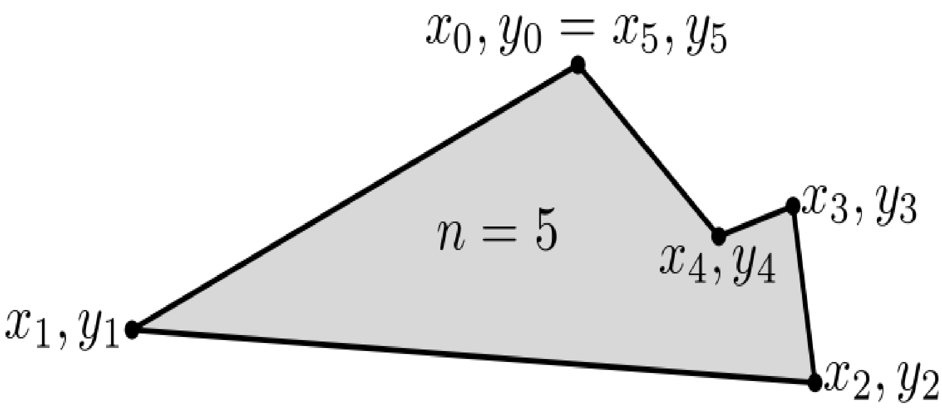


Figure 4.10: Shape of the irregular [polygon]

(a) Image with single concepts with Semantic Intensity (SI) value for an object car in the sample image

(b) Image with multi-concepts

Figure 4.11: (a) (b) Sample images related to single and multi-concepts

The semantic intensity (SI) for the given concept can be calculated on the basis of irregular polygon, as the polygon represents the edges of the objects in the image, while each object has a specific name in the image. We believe that, area of each object in a given image, with respect to the size of the image represent semantic intensity, the greater the semantic intensity value, greater will be the concept dominancy in the image and vice versa. The images that a single concepts are simple to understand for high level semantics, another words they are easy to tag with a single description. While the images that have many objects and have more concepts tag are a bit difficult to comprehend with a single semantic description. The Figure 4.11(a) shows an image with a single concept "car", which can be easily describe semantically, while the image in Figure 4.11(b), have more than one objects and concept tags, the images like this need more details to describe them semantically. The equation for semantic intensity (SI) calculation is as,

$$SI = \frac{A_{poly}}{I_s} \tag{4.5}$$

Where $I_s = h * w$, represents size of the image. The Figure 4.11(a) shows the SI of the object car in the image based on the value of the polygon.

Now the equation (5.2) becomes

$$C' = \cup_{j=1}^{m}(\cup_{i=1}^{n}(t', SI)_i)_j \tag{4.6}$$

The algorithm for the semantic intensity (SI) calculation is as under.

Propose Algorithm 4.1: Semantic Intensity Calculation

Input: L → $C' = \cup_{j=1}^{m}(\cup_{i=1}^{n} t_i')_j$

Output: *XML_file* → $C' = \cup_{j=1}^{m}(\cup_{i=1}^{n}(t', SI)_i)_j$

Method:

i → Length (L)

[X,Y] ← L.object(i).polygon
L.object(i).SI ← Polyarea(X,Y)/L.image(i).size

XML_file ← Struct2XML(L)

4.3.3 Image Annotation Similarity Matrix

It is well-known that similarity measure is the operation to compare the similarity of two sets, where each set can be analysed by some set relationships and set operations. The similarity matrix is based on the facts of the similarity among the images. The similarity matrix can be categorized as standard and weighted similarity matrix. The standard similarity matrix is based on the Boolean algebra, where each cell represents 0 or 1, the decision is either relevant or irrelevant. Relevant image is delineated by 1, while the irrelevant is represented by 0.

Figure 4.12: Standard Similarity Matrix, the Similarity measures for images close return a value of 1; However dissimilarity measures return a value of 0.

In the Figure 4.12, it is shown that using standard similarity matrix the image is either resides in the category of 0 or 1. There is no other possibility.

	I_1	I_2	I_3	I_4
I_1	1	0	0	1
I_2	0	1	1	0
I_3	0	1	1	0
I_4	1	0	0	1

Figure 4.13: Standard Similarity Matrix for a set of four images

The standard matrix fails to explain the degree of relevancy among the pair of images. In order to remove the bottleneck of the standard matrix, we have implemented the weighted matrix for our proposed module.

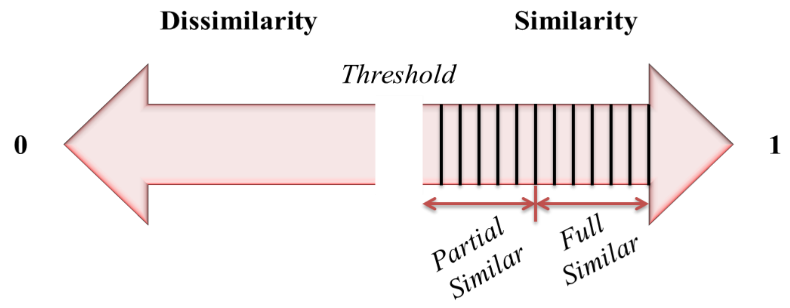


Figure 4.14: The Weighted matrix, it's not only find the relevant and irrelevant, but also find the degree of relevancy among the pair of images.

	I_1	I_2	I_3	I_4
I_1	1	0.81	0.40	0.69
I_2	0.81	1	0.54	0.85
I_3	0.40	0.54	1	0.65
I_4	0.69	0.85	0.65	1

Figure 4.15: Weighted Matrix for the four images

The weighted similarity matrix is further decompose into two sets, i.e. full similar (FS), partial similar (PS). The decision of the FS and PS are on the basis of the image relevancy i.e. images similarity. The similarity values among a pair of images fluctuated between 0 and 1, we have defined a threshold of 0.80 and above for FS and 0.50 for PS set. For example, the source image with a relevancy value greater or equal to 0.80 with any other images will be inserting into FS set, while value in a range of 0.79 and 0.50 will be a part of PS set. The values below 0.50 are considered to be an irrelevant images pair.

The similarity between two given images can be found by using concept tag with the objects and their semantic intensity (SI) values, the output of the *SIM(A,B)* for the two images set will be 1, if and only if, their concepts and SI values are equal, because it is not necessary that concepts matching in both images set will produce high result as same concepts might have different SI values in any given images. The output of the *SIM(A,B)* range from 0 to 1, where we keep 0.80 as a threshold value for the FS set, 0.50 value for PS set and values below 0.50 means no similarity and the images set are discarded straightaway. This process is continued until all of the images in the corpus are properly clusters (discussed next) into sets of FS and PS for all the images individually .i.e. for the first image we execute the process and obtain the FS and PS sets and then repeated for the second images and continue until all the images are properly clustered in FS and PS sets.

$$C'' = \bigcup_{j=1}^{m}(\bigcup_{i=1}^{n}(FS, PS, IS)_i)_j \tag{4.7}$$

Where FS, PS are the full, partial sets and IS is the image similarity. The algorithm for the image similarity is under.

Propose Algorithm 4.2: Image Similarity Calculation

Input: L → $C' = \bigcup_{j=1}^{m}(\bigcup_{i=1}^{n}(t', SI)_i)_j$

Output: *XML_file* → $C'' = \bigcup_{j=1}^{m}\left(\bigcup_{i=1}^{n}(FS,\ PS,\ IS)_i\right)_j$

Method:

SI1 ← 0
SI2 ← 0

$i \rightarrow Length\ (L)$

    j → 2: Length(L)

    IF (L.object(i).name = L.object(j).name) THEN

        SI1 ← SI1 + L.object(i).SI

        SI2 ← SI2 + L.object(j).SI

IF (SI1 ≥ SI2) THEN   IS ← SI1/SI2
ELSE IS ← SI2/SI1

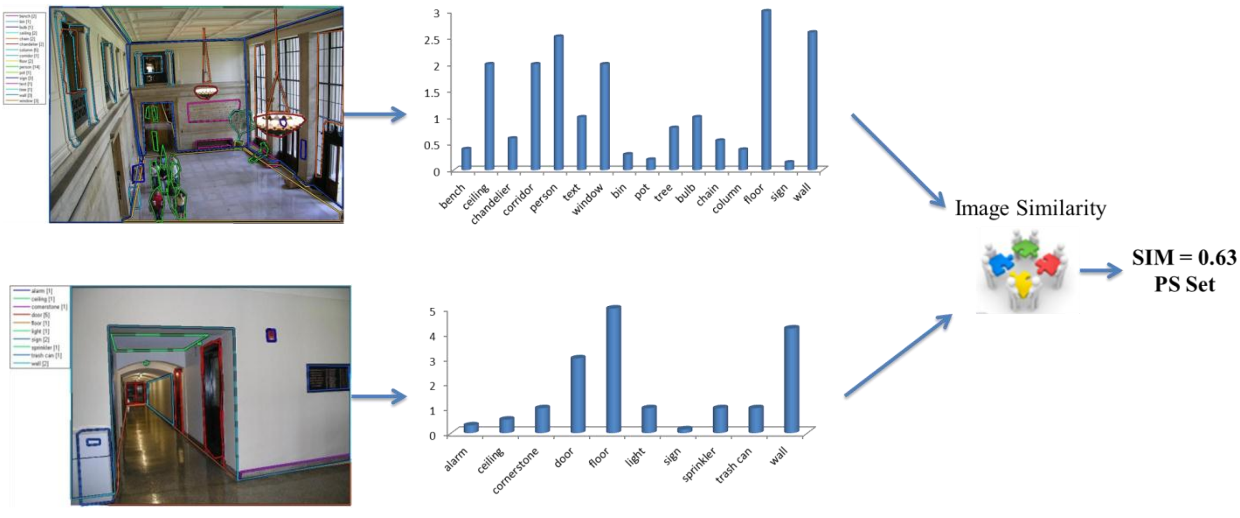


Figure 4.16: Image similarity measure on the basis of annotation.

4.3.4 Clustering the Similar Images

For achieving higher precision during retrieval, the high level semantic (HLS) propagation is the only possible solution, as many existing retrieval systems mechanism is based on the comparison of query image with rest of the images in the corpus, results in a high computational cost, especially when the corpus is too immense. Image archive categorization and group them into a clustering is an important step for effectively handling large image data sets. To solve the problem and to ease the process HLS propagation process for the LabelMe corpus, we use the categorization and clustering technique for each image, where a set of full similar (FS) and partial similar (PS) are prepared on the basis of image similarity using annotation. Image classification and grouping them into FS/PS sets are a means for high-level description of image content. The goal of making the FS/PS sets for each image is to find similar images with similar contents or they share same/partial semantics. As a result the mapping of HLS to the images sets will provide essential information about the image archive. Adding to this, each group of the category of the images will share the same information, while their annotation file will be maintain separately. The advantage of this will be benefit during the query process phase, where a computational cost of the query will be minimize in finding similar images and query will produce result in a smart way with high precision.

A variety of clustering techniques have been developed to group documents into topically-coherent. This can help users to browse through the search results, obtain an overview of their main topics/themes and help to limit the number of documents searched or browsed in order to find relevant documents (i.e. limit search to only those clusters likely to comprise relevant documents). Based on the literature survey the clustering can be categorize into the following three main types

4.3.4.1 Hierarchical Clustering

The Hierarchical clustering approach builds a hierarchy (a tree) where the nodes in the tree represent the clusters. This approach can be used in either a bottom up or top-down fashion creating a new level of clusters at each iteration.

A clustering algorithm can be agglomerative [Amadsun et al. 1988] or divisive [Choudhury et al. 1990]. Strategies for hierarchical clustering generally fall into two types:

- ***Agglomerative*****:** This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- ***Divisive:*** This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

4.3.4.2 Partitional Clustering

Using Partitional clustering means to partition the dataset into a number of parts (clusters). The number of parts is defined beforehand and the algorithms refine these parts at each iteration to improve these parts. The algorithms stop when they have converged or a number of iterations are done. An example of a partitional clustering algorithm is the original *k*-means algorithms. The unmodified version of Bisecting *k*-means can also be seen as a partitional clustering algorithm.

4.3.4.3 Spectral Clustering

The last approach is Spectral clustering. The spectral clustering algorithms usually use dimensionality reduction techniques such as Singular value decomposition or Non-negative matrix factorization to reduce the dimensionality of the datasets so that they are easier to work with. Clustering of the dataset is then performed on the dimension reduced set. Example of spectral algorithms is *latent semantic indexing* and *probabilistic latent semantic indexing*.

Recent advances in data mining allow for exploiting patterns (e.g., a set of binary attributes) as primary means for clustering large collections of data. Another significant issue in image clustering is that images with similar semantics may not fall in one cluster as image clustering is performed based on image low-level features. Many approaches have been proposed to reduce the gap between high-level image semantics and low-level image features and improve the clusters by applying image segmentation techniques on region-based features and clustering image segments instead of original images. Since all image low-level features cannot capture high-level semantic concepts, most retrieval methods have tried to find an optimum set of feature weights to model the user's perception based on image features (feature weighting).

We have tried to make the cluster on the basis of image semantic similarity value. There is no specific criterion for making the cluster as all of the images are already annotated and we need only to group the images on the basis of concepts tag with the objects and their SI values in the annotation. We used a widely used hypergraph partitioning algorithm, called hMETIS [Karypis et al. 1996], to partition the feature hypergraph. hMETIS produces *"balanced k-way"* partitions where *k*, the number of partitions, is specified in advance.

a) Hyper Graph

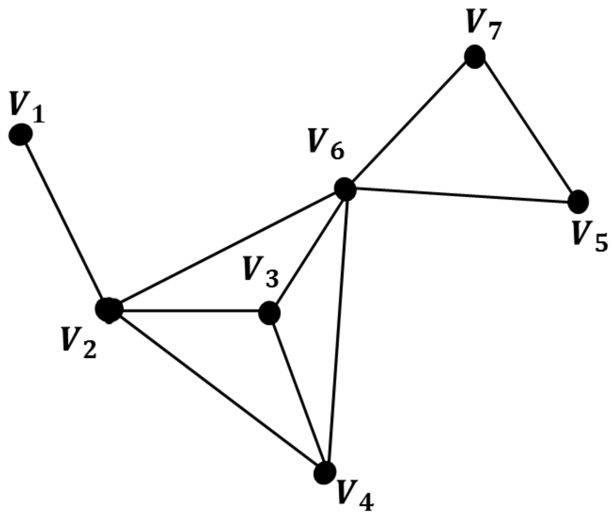
Hypergraphs have proven useful in data mining and high-dimensional document clustering problems [Han et al. 1998], [Han et al. 1997]. Hyper graph can be define as,

"*A pair of sets* H = (V, E). V *is the set of* vertices *of the hypergraph and* E *is the set of* hyperedges *of the hypergraph. Each hyperedge in a hypergraph is a non-empty subset of* V, *the size of this subset is called the hyperedge's* degree. *A* weighted hypergraph *has non-negative numeric weights associated with each vertex, each hyperedge, or both"*

In a typical hyper graph, each vertex represents a dimension and each hyperedge represents an affinity (or relationship) between two or more dimensions represented by the corresponding vertices. Weights assigned to vertices indicate importance of these vertices and weights assigned to hyperedges indicate the strength of the relationship between dimensions represented by the vertices connected by a hyperedge.

	E_1	E_2	E_3
V_1	1	0	0
V_2	1	0	1
V_3	0	0	1
V_4	0	0	1
V_5	0	1	0
V_6	0	1	0
V_7	1	0	1

(a)



(b)

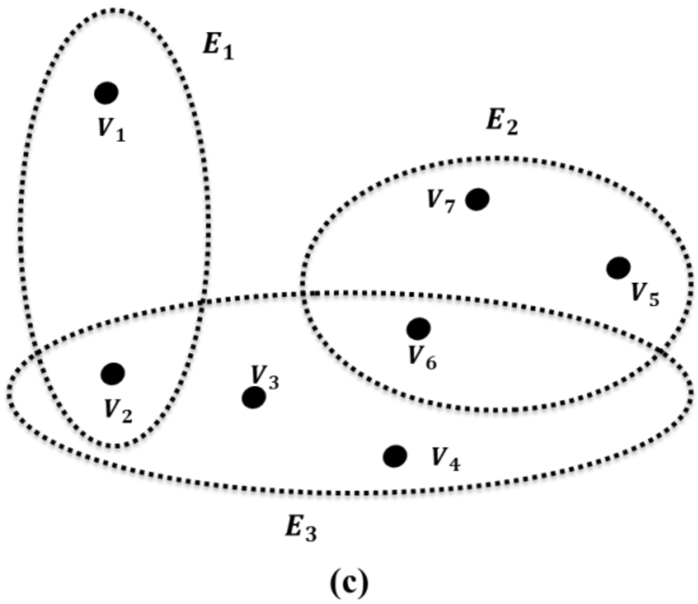


*Figure 4.17: Hypergraph[*Book -2*] vs. simple graph. (a) Tabular representation, where set E = {e1; e2; e3} and an images set V = {v1; v2; v3; v4; v5; v6; v7}. (b) An undirected graph in which two images are joined together by an edge if there is at least one feature in common. (c) A hypergraph which completely illustrates the complex relationships among images.*

Let $\mathrm{H} = (\mathcal{V}, \mathcal{E})$ is the weighed hyper graph, where

$\mathcal{V} = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{n}\}$: a finite set of images.

$\mathcal{E} = \{\mathrm{E}_1, \mathrm{E}_2, \dots, \mathrm{E}_\mathrm{m}\}$: a properties of subsets of $\mathcal{V}$

$\mathrm{E}_\mathrm{j} \neq \phi, \mathrm{j} = 1, \dots, \mathrm{m}$

$\cup_\mathrm{j} \mathrm{E}_\mathrm{j} = \mathcal{V}$

$\omega = \text{weight}$

The process of computing a coarser hyper graph from an input hyper graph by merging vertices into larger groups of vertices called *clusters.* The weight of each cluster will be the sum of the weights of its vertices, or simply the number of vertices if they have no weights. Based on the observation that using association rules directly for clustering may result in clusters that are too granular, Han et al. [Han et al. 1997] proposed an approach to cluster transactions using association rule hypergraphs. A *hypergraph* is similar to a graph except that each edge, called a *hyperedge*, can connect two or more vertices. In order to generate a hypergraph from a set of association rules, each unique item that exists in the set is assigned to a unique vertex in the graph.

b) Discussion

Let $\mathcal{V} \leftarrow$ (A, B, C, D, E, F) set of annotated images having a list of concepts.

$A \leftarrow \{bench, text, ceiling, person, floor, tree, wall, light\}$

$B \leftarrow \{ceiling, door, floor, light, wall, alarm, text, board, stair\ rail\}$

$C \leftarrow \{board, ceiling, door, floor, light, smoke\ detector, wall\}$

$D \leftarrow \{car, chimney, door, house, person, window, tree, side\ walk, sky\}$

$E \leftarrow \{building, car, chimney, road, side\ walk, tree, window, sky\}$

$F \leftarrow \{person, building, car, road, side\ walk, tree, window, sky\}$

For simplicity, we will assign variables to each of the unique concept tag with the images. The lists of all the concepts with their variables are $x_1 \leftarrow alarm, x_2 \leftarrow bench, x_3 \leftarrow board, x_4 \leftarrow building, x_5 \leftarrow car, x_6 \leftarrow ceiling, x_7 \leftarrow chimney, x_8 \leftarrow door, x_9 \leftarrow floor, x_{10} \leftarrow house, x_{11} \leftarrow light, x_{12} \leftarrow person, x_{13} \leftarrow road, x_{14} \leftarrow side\ walk, x_{15} \leftarrow sky, x_{16} \leftarrow smoke\ detector, x_{17} \leftarrow stair\ rail, x_{18} \leftarrow text, x_{19} \leftarrow tree, x_{20} \leftarrow wall, x_{21} \leftarrow window$

So the edges of the vertices are calculating using the intersection

$E_1 \leftarrow (A \cap B) \leftarrow (light, floor, ceiling, wall, text) \leftarrow (x_6, x_9, x_{11}, x_{18}, x_{20})$

$E_2 \leftarrow (A \cap C) \leftarrow (floor, light, wall) \leftarrow (x_9, x_{11}, x_{20})$

$E_3 \leftarrow (A \cap D) \leftarrow (person) \leftarrow (x_{12})$

$E_4 \leftarrow (A \cap E) \leftarrow (tree) \leftarrow (x_{19})$

$E_5 \leftarrow (A \cap F) \leftarrow (person, tree) \leftarrow (x_{12}, x_{19})$

$E_6 \leftarrow (B \cap C) \leftarrow (board, ceiling, door, floor, wall) \leftarrow (x_3, x_6, x_8, x_9, x_{20})$

$E_7 \leftarrow (B \cap D) \leftarrow (door) \leftarrow (x_8)$

$E_8 \leftarrow (B \cap E) \leftarrow (door) \leftarrow (x_8)$

$E_9 \leftarrow (B \cap F) \leftarrow \phi$

$E_{10} \leftarrow (C \cap D) \leftarrow (door) \leftarrow (x_8)$

$E_{11} \leftarrow (C \cap E) \leftarrow \phi$

$E_{12} \leftarrow (C \cap F) \leftarrow \phi$

$E_{13} \leftarrow (D \cap E) \leftarrow (car, chimney, tree, sky) \leftarrow (x_5, x_7, x_{15}, x_{19})$

$E_{14} \leftarrow (D \cap F) \leftarrow (car, tree, sky) \leftarrow (x_5, x_{15}, x_{19})$

$E_{15} \leftarrow (E \cap F) \leftarrow (car, chimney, building, road, side\ walk, tree, window, sky)$

$\leftarrow (x_4, x_5, x_7, x_{13}, x_{14}, x_{15}, x_{19}, x_{21})$

Weight of the images are calculated on basis of image similarity (discussed above), the following are the weights of the different edges of the vertices.

$W_1(A \leftrightarrow B) \leftarrow 0.81,\ W_2(A \leftrightarrow C) \leftarrow 0.56, W_3(A \leftrightarrow D) \leftarrow 0.18,$

$W_4(A \leftrightarrow E) \leftarrow 0.10, W_5(A \leftrightarrow F) \leftarrow 0.23, W_6(B \leftrightarrow C) \leftarrow 0.87,$

$W_7(B \leftrightarrow D) \leftarrow 0.12, W_8(B \leftrightarrow E) \leftarrow 0.12, W_9(B \leftrightarrow F) \leftarrow \phi,$

$W_{10}(C \leftrightarrow D) \leftarrow 0.18, W_{11}(C \leftrightarrow E) \leftarrow \phi, W_{12}(C \leftrightarrow F) \leftarrow \phi,$

$W_{13}(D \leftrightarrow E) \leftarrow 0.72, W_{14}(D \leftrightarrow F) \leftarrow 0.59, W_{15}(E \leftrightarrow F) \leftarrow 0.98$

The hypergraph representations of the above images are shown in figure 4.18, while the empty and set that having single elements are ignore.

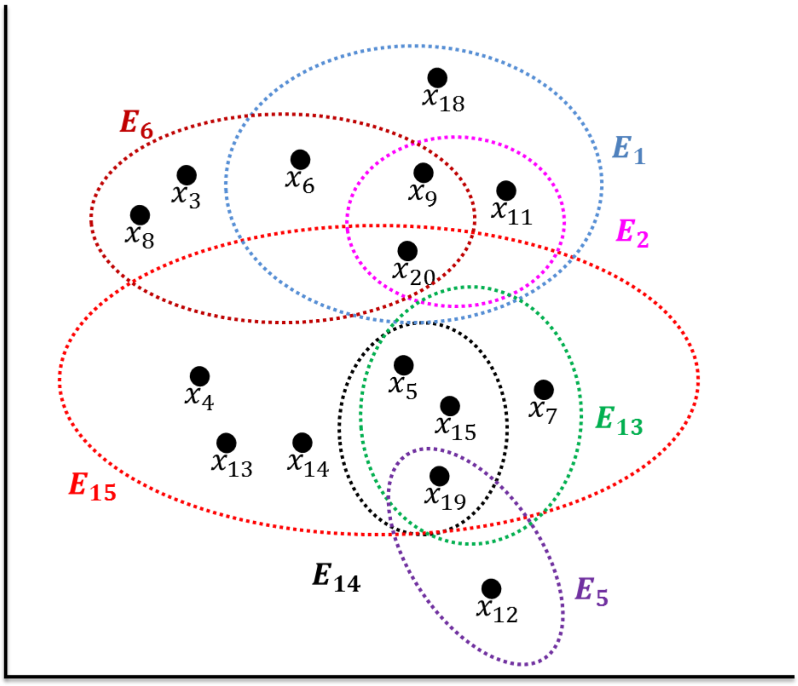


Figure 4.18: *The clustering of the common features among the images, where edges of the vertices (images) that share the common concepts are grouped into one cluster using the hypergraph hMETIS [Karypis et al. 1996] algorithm.*

The dendogram representation of the proposed hypergraph approach for clustering the images is shown in Figure 4.19.

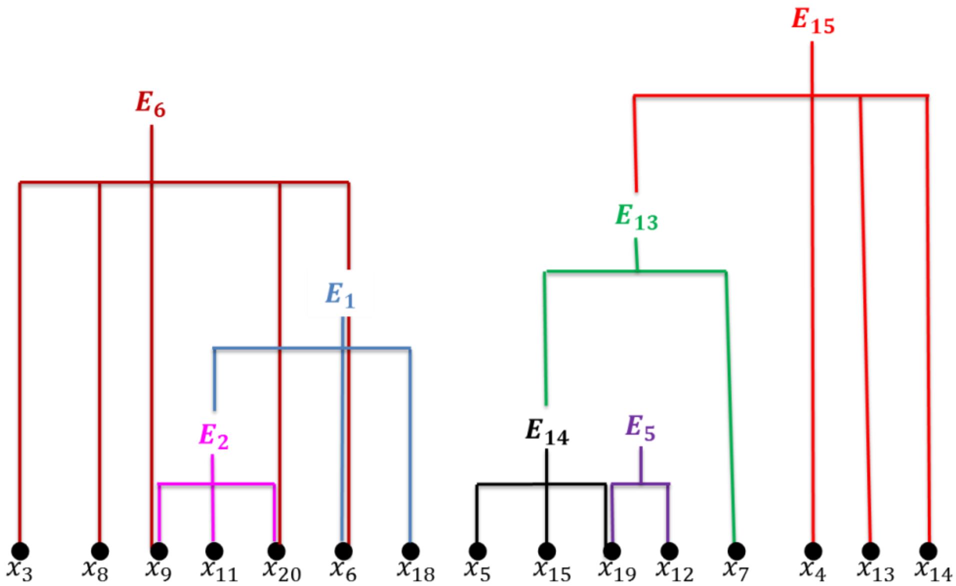


Figure 4.19: Dendogram illustration of the proposed concept space for the randomly selected 6 images from the LabelMe images corpus.

c) Example

Let we have a set of six images (A, B, C, D, E, F) in the corpus, while their similarities values (i.e. weightages) among them are described above. The logical partitioning of the images into a set of FS and PS are shown in the Figure 4.20, where each image have other images in the FS and PS sets, while images B and E have PS $\leftarrow \phi$. The images with FS $\leftarrow \phi$, have a unique concepts among the others and that need special attention during the HLS propagation process, although they get the high level semantic description but that partially depicts the entire semantics and not fully understandable. So during the HLS process, the images like this are describe separately. In the next section, we will discuss how high level semantic (HLS) propagation work. The XML format of the annotation for the image similarity is shown in Figure 4.21.

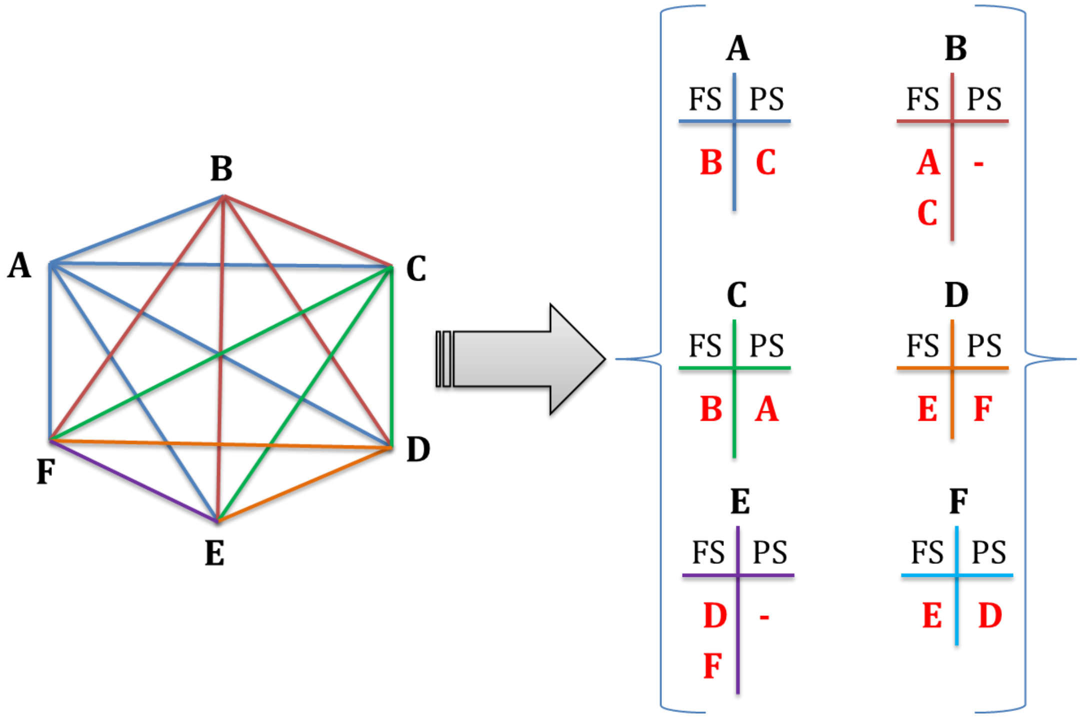


Figure 4.20: The example for the images similarity and clustering set mechanism among the four images set (A, B, C, D, E, F)

```
<Annotation>
<folder>p1010844.jpg</folder>
<filename> 05june05_static_indoor </filename>
<source>
        <sourceImage>The MIT-CSAIL database of objects and scenes</sourceImage>
        <sourceAnnotation> LabelMe Webtool</sourceAnnotation>
</source>
<SemDescp> Indoor view of the building where light, corridor view and stair are mentioned  </SemDescp>
<FS>
        <image>
                <ID> 1 </ID>
                <folder> 05june05_static_indoor </folder>
                <filename> p1010846.jpg </filename>
                <SemDescp> Indoor view of the building where alarm, bin, and corridor view are mentioned</SemDescp>
                <IS> 0.93 </IS>
        </image>
</FS>
<PS>
        <image>
                <ID> 1 </ID>
                <folder> 05june05_static_indoor  </folder>
                <filename> p1010843.jpg </filename>
                <SemDescp> Indoor view of the building, where people, bench, light are mentioned </SemDescp>
                <IS> 0.67 </IS>
        </image>
</PS>
</Annotation>
```

Figure 4.21: XML format of the image similarity annotation handling

4.3.5 HLS Propagation

The high level semantic propagation is the process of assigning semantic annotation description to the images in the corpus, while the cluster mechanism discussed in the previous sections provide an environment where a single effort for the annotation can be easily propagate through rest of the images via the FS and PS sets along with their similarity values. The idea of keeping similarity values during the HLS propagation process is to keep the ration of the relevancy of the semantic description of the original and images in the sets, this will not only benefit us to maintain the cross checking of the HLS among the images but will also provide an opportunity to cross-check the description among both images for consistency. Those images having FS and PS sets with either $FS \leftarrow \emptyset$, $PS \leftarrow \emptyset$ or their both sets $FS \wedge PS \leftarrow \emptyset$ are annotated manually. The high level semantic queries on this type of HLS annotation rank the output of the images on the basis of their SIM values either from FS or PS set or their combination. The algorithm for the HLS propagation is under

---

Propose Algorithm 4.3: High Level Semantic Propagation

Input: L → $C^{"} = \bigcup_{j=1}^{m} \left(\bigcup_{i=1}^{n} (FS,\ PS,\ IS)_i \right)_j$

Output: *XML_file → XML_file with HLS*

Method:

*i ← Length(*L.FS*)*

L.FS(i).SemDescp ← L.SemDescp

*j ← Length(*L.PS*)*

L.PS(j).SemDescp ← L.SemDescp

4.4 Experiments and Evaluation

We used the [LabelMe] dataset for the experiments, which contains total of 181, 932 images with 56946 annotated images, 352475 annotated objects and total of 12126 classes. It was difficult for us to test the proposed system on all of the images, so we only select 500 images randomly.

In case of HSL, we achieve good results in FS and PS sets, the Figure 4.22 shows comparison of FS and PS set for the three randomly selected HLS example.

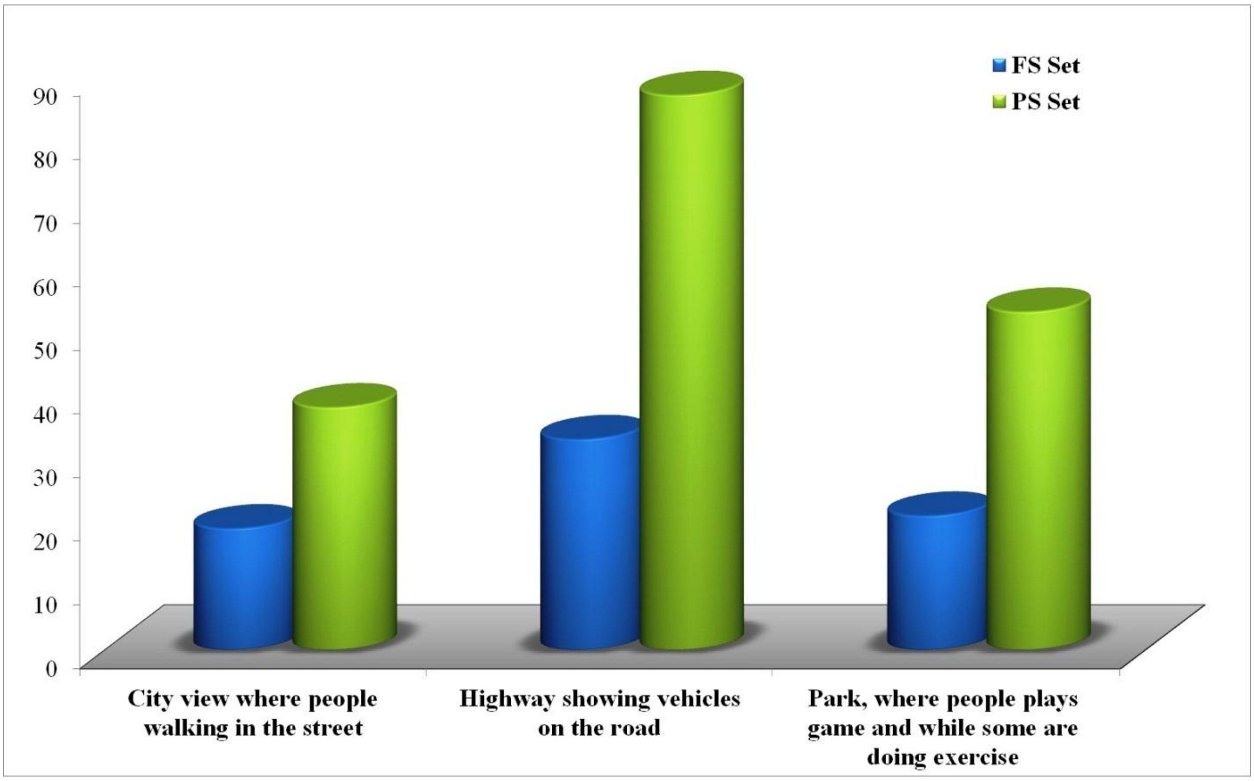


Figure 4.22: Example of the HSL annotation on Full Similar (FS) and Partial Similar (PS) sets

The Figure 4.22 shows the proportion between the FS and PS sets. The basic intension of categorizing the images into FS and PS sets for minimizing the human intervention and automatic the process of high level semantic description of the images. The basic idea for the categorization of the images into Full Similar and Partial Similar sets are on the basis of the novel concepts, i.e. Semantic Intensity (SI) of the different concepts within the single image.

It is a well-known fact that image is the combination of different objects and different combination of these objects constitutes different semantics meanings. Some of the concepts within the image are more dominant than the others. The proposed technique intents to categorize the images on the basis of matching the concepts tags with the images and their semantic intensity (see section 4.3.2). In the Figure 4.22, the number of the PS set have high value than that of FS, which is due to the facts, that it is very rare to agree that two images fully share the same semantics. For instance, the two images may contain the similar object combination but different semantic idea, like the images of the simple high level concept, i.e. *car park* and the *street* may contain the objects like *tree, road, people, car, building, sky,* etc. Even though both the concepts contain the same object constitution but the difference is the dominancy level of the objects. In the *street* view the object like the *car* is less dominant, while for the images contain the concept *car park* have the *car* object more dominant than other concepts like *people*, *building*, etc., which are more dominant in *street* view. The traditional system that based on the primitive feature extraction and object recognition and matching techniques flunks to differentiate among the images of both these concepts. We attempt to remove this bottleneck of the traditional system by exploiting the semantic intensity for differentiating the images of street with the car park. This is the reason why the full similarity between the images is rare. While partial semantics is possible due to the dynamics semantics of the images, i.e. in case of PS sets the image gets more than one HLS description representing their dynamics in semantics.

Information science has developed many different criteria and standards for the evaluation e.g. effectiveness, efficiency, usability, satisfaction, cost benefit, coverage, time lag, presentation and user effort, etc. Among all these evaluation technique precision which is related to the specificity and recall which are related to the exhaustively are the well accepted methods. As used by the previous researchers, the quality of the image annotation in terms of high level semantics can be measured through the precision and recall. Per-image precision and recall are calculated on the basis of a single test image taking from the corpus prepared for the high level semantic propagation. For each test image, precision is defined as the ratio of the number of semantic description that are correctly predicted to the total number of possible semantic description prediction tag with the image in the cluster set, and recall is the ratio of the number of semantic description that are correctly predicted to the number of semantic description in the cluster sets. Mathematically, they are calculated as follows

$$Per\ Image\ Recall = \frac{\#\ correclty\ semantic\ description}{\#\ semantic\ description\ of\ images\ in\ cluster\ set} \quad \ldots \quad (4.8)$$

$$Per\ Image\ Precision = \frac{\#\ correclty\ semantic\ description}{\#\ total\ semantic\ description\ tags\ with\ images\ in\ cluster} \quad \ldots \quad (4.9)$$

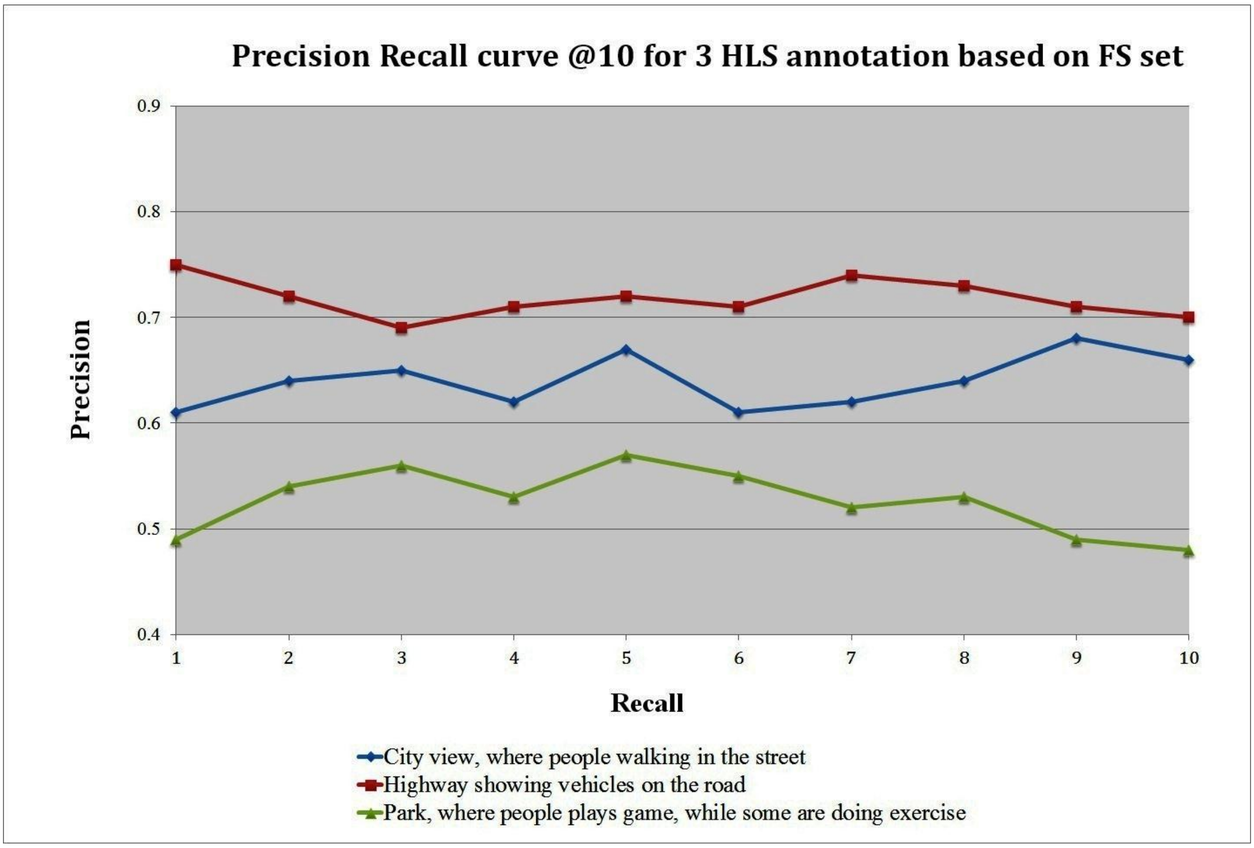


Figure 4.23: Precision and recall in term of HLS description for the FS set of 10 sample images.

For high level semantic annotation propagation, for the validation and verification of the effectiveness of the proposed framework, we applied queries on the corpus and check the results. The proposed techniques achieve a noticeable improvement in terms of precision and recall. The Figure 4.23 shows the precision and recall of the top 10 query results for the three randomly selected HLS annotation as a query. The three HLS annotation is (1) City view, where people walking in the street. (2) Highway showing vehicles on the road. (3) Park, where people plays game, while some are doing exercise. The precision recall curve depicts a tremendous improvement in terms of specificity and exhaustively based on the FS set of the images. There is a variation among the three selected semantically enriched high level conceptual queries. This variation is due to the fact that, as with the increase in the complexity sometimes, the precision of the system decreases, and it is difficult to deal with. The high level semantic concepts like *Park* which itself a heteronym (words that have same spelling with different meaning). *Park* shares two concepts, i.e. *car park* and *recreation park*, dealing with such types of queries are very difficult. While in the Figure 4.23, the high level semantic concept *Park* also contains the concepts *people* and *game*, so it directs towards the *recreation park*. However, still in most of the circumstances the precision of such types of queries are less. The mean average precision for the queries based on the full similar set are, for the *City view, where people walking in the street* query is 0.64, for *Highway showing vehicles on the road* mean average precision is 0.72, while for the query *Park*, *where people play games, while some are doing exercise* mean average is 0.53.

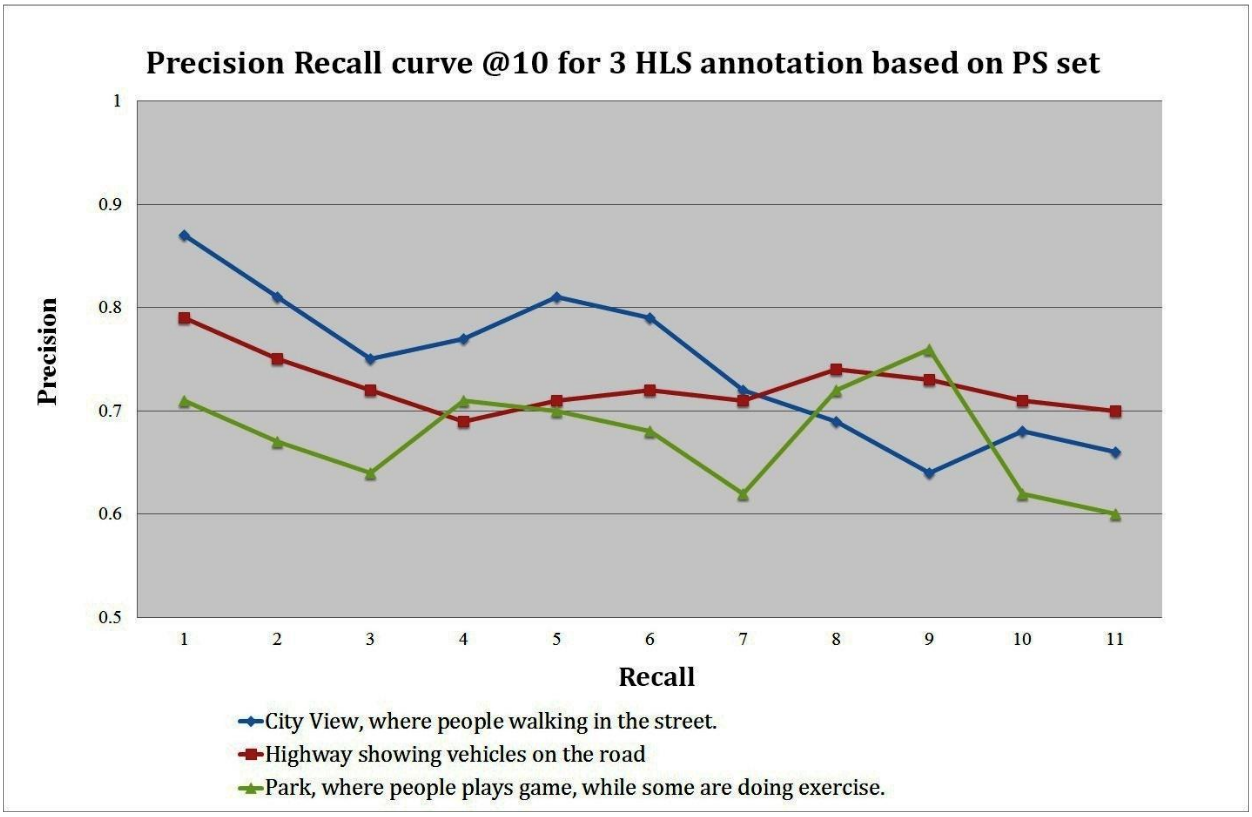


Figure 4.24: Precision and recall in term of HLS description for the PS set of 10 sample images.

The Figure 4.24 shows the precision recall curve @10 for the same three HLS annotation that was used for FS was also used for PS set. The curve for the PS is increased as compared to the FS set (Figure 4.23) due to the fact, the chances for the partial semantic sharing is high among the images as compared to the full similarity. The mean average precision for the City view, where people walking in the street query is 0.74, for Highway showing vehicles on the road mean average precision is 0.73, while for the query Park, where people play games, while some are doing exercise mean average is 0.68.

4.5 Chapter Summary

The focus on this chapter is on the process of manual annotation for the object annotated image datasets. Where we present a novel framework for the HLS support, this kind of work is a unique approach to date for the HLS annotation for a large scale images corpus. This framework can be easily turned into automatic by integrated an automatic object detector and recognizer. The flow of work of the framework is based on the cluster set of full similar (FS) and partial similar (PS) are prepared for each of the images individually by using the image similarity mechanism, where a define threshold of 0.80 and 0.50 are declared for FS and PS sets. High Level Semantic description is then propagated by assigning them to one image and the system automatically spread it out that to all the images in FS and PS sets. This technique abbreviates the effort for the manual annotation and produces high semantic accuracy in terms of precision for large pool of image data sets. It stipulates a rich inside of the image in term semantics rather than the contents of the image. The experiments were investigated on the random selected portion of the LabelMe data sets. Improvements have been made in terms of semantic accuracy, effort and precision.

Chapter 05

Annotation Enhancement & Refinement for Video

"If I have seen further it is because I have stood on the shoulders of giants"
Merton, 1993

At the beginning of this millennium in the course of rapid societal transformation processes another new development in technology enters and consolidates an important position in the video business: The computers as multimedia equipment and other devices are going to change the handling of videos completely. The need for intelligent mining and management tools, for hugely increasing amount of video collections available, became crucial. This motivated the work on Video Understanding applications, like semantic video annotation, rating, indexing and retrieval. Work in this area aims to fill the “Semantic gap”, which is the difference between low-level visual features and human’s perception. A number of approaches try to establish a semantic representation of visual data in textual form to tackle this issue. For achieving this aim, these approaches either build a domain specific “Ontology”, which refers to the theoretical representation model in knowledge systems [1], or focus on the content by applying the image analysis techniques.

In this chapter, we extend the previous work from images to video domain by applying the concept enhancement and refinement techniques to the LabelMe videos datasets. From the experiments on the specified datasets, we achieve a noticeable improvement in term of concept diversity, enrichment ration and retrieval degree.

The rest of the chapter is organized as follows. In the next section, a brief introduction about the video is presented along with the existing trends of the market. The section 5.3 is dedicated to describe the video document and their representation, where different element like shot, scene and key-frame of the video analysis are under discussion. Section 5.4 covers the state-of-the-art for annotation in the video domain. Section 5.5 introduce the proposed framework for video, while section 5.6 emphasis on the evaluation measure of the proposed work.

5.1 Introduction

Media analysis for video indexing is spotting an increasing impact of statistical techniques. Examples of these appearances include the use of generative models as well as discriminant techniques for video summarization, structuring, indexing, retrieval and classification. There is increasing emphasis on diminishing the amount of supervision and user interaction required to build and make use of the semantic models. Because, interacting with video in particular and multimedia data in general, involves more than connecting with data banks and providing data via networks to customers' homes or offices. We still have limited tools and applications to organize, manage and describe video data. A simplified multimedia information retrieval application is composed by a multimedia database, analysis algorithms, a description database, and a user interface application. Analysis algorithms extract the low-level signature from multimedia and store them as descriptions of that content. A user then deploys these indexing descriptions in order to search the multimedia database. A typical semantic multimedia information retrieval framework is shown in Figure 5.1 differs eminently from traditional retrieval applications on the low-level analysis algorithms; its algorithms are responsible for extracting semantic information used to index multimedia content by its semantic. Multimedia content can be indexed in many ways, and each index can refer to different modalities and/or parts of the multimedia piece. Multimedia content is composed of the visual track, sound track, speech track, and text. All these modalities are arranged temporally to provide a meaningful way to transmit information and/or entertainment. Manually forming video content description is time consuming and therefore, more costly, to the point that it's almost impossible. Moreover, when available, it's subjective, inaccurate, and incomplete.

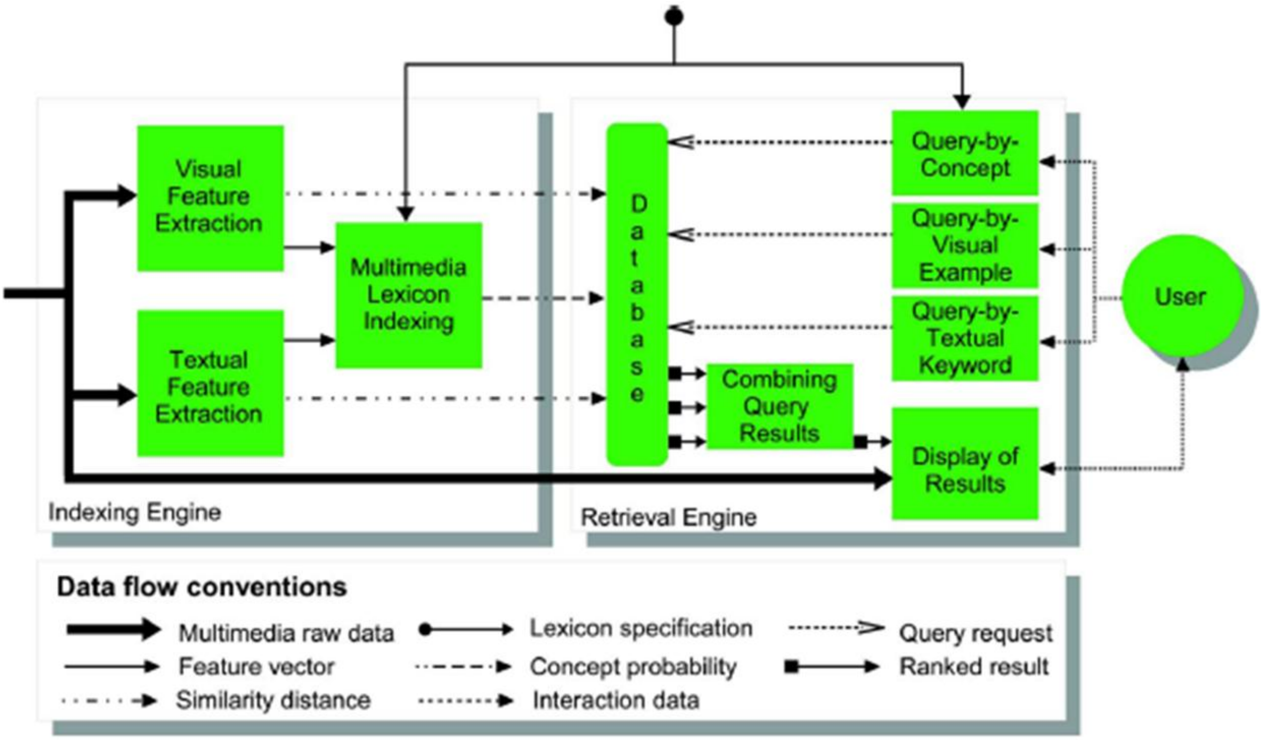


Figure 5.1: Video retrieval system framework [Snoek et al., 2007]

The increase in interest of managing multimedia collections efficiently and effectively has created new research importance that arises as a combination of information extraction, digital libraries, information retrieval and multimedia understanding. This growing interest has resulted in the creation of a video retrieval track in TREC conference series in parallel with the text retrieval track (TRECVID, 2010).

There is a sizeable amount of work effectively dealing with the description of audio-visual media, but most of it focuses on two genres: news and sports broadcasts. One reason for this fact is the commercial relevancy, as segments of sports and particularly news content are frequently reused after their production (e.g. when the aeroplane crashed, hours of ad hoc programs had to be filled with archive documentation material of the aeroplane, because only a one minute sequence from the actual crash existed) and initial airing, so that they are valuable assets for broadcasters. Segments from feature films are hardly reused in other contexts, so that a detailed annotation is commercially not interesting. Further, compared to feature films, news and sports broadcasts have very clear dramaturgical structures, which makes the automation of segmentation (for example of news stories) more feasible.

5.2 Video Structure and Representation for Annotation

A video is a structure of still images, played with by an audio stream. Classical digital video standards are the MPEG-1 and MPEG-2 formats. They were released by the Motion Pictures Expert Group (MPEG), the driving force in the development of compressed digital video formats. MPEG-1 videos are often compared to old fashioned VCR recordings. The newer MPEG-2 video format is used to encode videos in DVD quality. Coupled with the increased power of computing, manipulation of digital videos is now increasing. The way video documents are temporally structured can be distinguished in two levels: semantic and syntactic structure.

At the syntactical level, the video is segmented into shots (visual or audio) that form a uniform segment (e.g., visually similar frames); representative key-frames are extracted from each shot, and scenes group neighbouring similar shots into a single segment. The segmentation of video into its syntactic structure of video has been studied widely [Brunelli, et al. 1999; Wang et al. 2000].

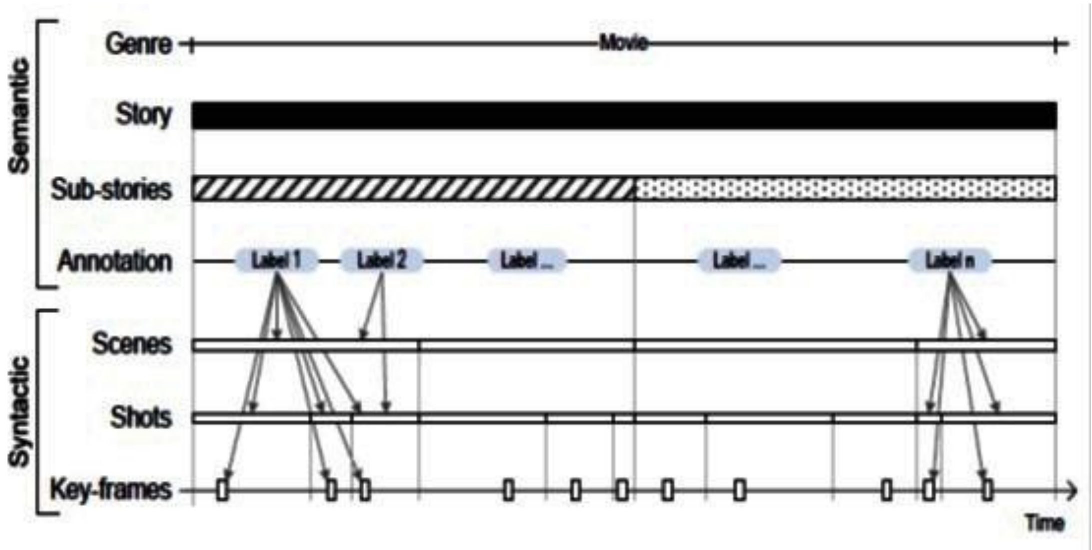


Figure 5.2: Syntactic and semantic structure of video [Magalhaes et al. 2007]

At the semantic level, annotations of the key-frames and shots with a set of labels indicate the presence of semantic entities, their relations, and attributes (agent, object, event, concept, state, place, and time (see [Benitez et al., 2002], for details). Further analysis allows the discovery of logical sub-units (e.g., substory or subnarrative), logical units (e.g., a movie), and genres. A recent review of multimedia semantic indexing has been published by [Snoek, et al. 2005].

To navigation in a video, it is necessary to break up the data into structured elements. In the case of video, these elements are shots, scenes and key-frames. A short discussion about the syntactic structure of the video is as.

The atomic unit of access to video content is often considered to be the video shot. Monaco [Grand Prix, 2009] defines a shot as a part of the video that results from one continuous recording by a single camera. A scene is composed of a number of shots, while a television broadcast consists of a collection of scenes. The gap between two shots is called a shot boundary. According to [Zhang et al. 1997], there are mainly four different types of common shot boundaries within shots:

- *A cut*: It is a hard boundary or clear cut which appears by a complete shot over a span of two serial frames. It is mainly used in live transmissions.

- *A fade*: Two different kinds of fades are used: The fade-in and the fade-out. The fade-out emerges when the image fades to a black screen or a dot. The fade-in appears when the image is displayed from a black image. Both effects last a few frames.

- *A dissolve*: It is a synchronous occurrence of a fade-in and a fade-out. The two effects are layered for a fixed period of time e.g. 0.5 seconds (12 frames). It is mainly used in live in-studio transmissions.

- *A wipe*: This is a virtual line going across the screen clearing the old scene and displaying a new scene. It also occurs over more frames. It is commonly used in films such as *Star Wars* and TV shows.

As these effects exist, shot boundary detection is a non-trivial task. It is not known before, when these effects will appear. There have been a number of diverse approaches to handle various shot boundaries, including calculating pixel differences between neighbouring frames, macro-block comparison from MPEG-encoding, comparison of neighbouring frames using colour-histograms and the comparison of edges in frames. All approaches work well for different transition types but cannot be used for every shot boundary. Frame comparison based on colours for instance works fine on cuts but does not detect dissolves or fades. Edge detection works effectively in wipe and dissolves detection. However, separating videos into different shots is not the best solution as the context of a shot is not often clear. Very often, a shot is only understandable when it is played in its context. A shot e.g. showing a public square full of people waving flags shows nothing more than a crowded square. Seen in its context, this crowd might be celebrating a victory of their favourite football team, celebrating the national day or demonstrating or protesting against something. Keeping the context of a video part is important for understanding it.

It is time consuming to browse through all video sections to find the relevant part [Girgensohn et al., 2005]. As a fundamental step of video indexing, scene cut detection algorithms have been widely studied to divide video streams into elemental units (i.e. shots). Low-level features such as colour, edge and motion have been proved to be appropriate for the detection of temporal changes such as camera breaks and transitions [Meng, et al. 1996, Zhang et al., et al. 1995]. Based on temporal segmentation, video data can be efficiently represented in an abstracted or summarized way. Many technologies have been developed to index segmented video shots.

One habitual approach that has been used in many systems is prior to selecting one or more key frames (i.e. representative frames) for each video shot, and then exercise image features such as colour, texture and shape to index these key frames. How to choose and organize key frames are the major issues here. Besides simple sampling methods, advanced algorithms have been developed to use colour variances, camera motions, embedded texts and human faces [Wang, et al., et al. 1996] to select frames that convey the most significant information of a video shot.

Exploitation only key frames for indexing ignore motion information included in video shots. Moreover, as the videos are broken into individual shots, events and temporal relationships among successive shots are not explored. Shot and scene semantic analysis puts forward the time dimension to the problem at hand. The time dimension includes temporal frames, resulting in additionally information to help the analysis. To enable search for events and actions, a number of methods have been proposed to include motion and temporal information into video content models. In [Chang, et al., et al. 1987, Bimbo, et al., et al.

1995], symbolic descriptions are used to represent temporal relationships (e.g., before, after, etc.) and to enable match and query of such temporal structures. Motion estimation, spatial-temporal logics, object segmentation and tracking are some key techniques that have been applied in such modelling processes.

Visual features comprise small-scale semantic information, and in many circumstances, are not adequate or comfortable for users to look for desired videos. High-level abstractions and summarizations, such as story, scene or action, allow users to search and browse videos at a more effective and intuitive level. For example, a news story from CNN is broken down into a hierarchy of segments, stories and then individual shots [Zhong, et al., et al. 1996]. This hierarchical structure allows a multiple layer abstraction that can be used to aid users navigate through the lengthy video program. In addition to detecting temporal structure, efforts have also been made to extract semantic segments from video shots.

In [Zhang, et al., et al. 1994], a spatial structural model is used to detect anchor-person scenes. A long news program is then broken into stories based on anchor-person scenes. In [Yeung et al. 1996], the scene transition graph is used to capture both the content and temporal flow of videos. It is reported to be efficient to detect actions, story and dialogues units.

In general, unlike elementary video shots that can be described based on low-level features, high-level entities like story or scene are difficult to automatically extract based on only low-level visual features. As observed in [Yeung et al. 1996], to properly group or classify video shots, more complicated domain models need to be built based on intermediate or high-level representations, such as regions or objects. In recent studies, several emerging video representation frameworks such as MPEG-4 and MPEG-7 have also adopted similar object-oriented models [MPEG-4, 1996, MPEG-7, 2000].

In conclusion, while improvement has been made in the area of video summarization and indexing, many stimulating issues remain to be solved. Thus, more advanced video analysing techniques are demanded to build effective and efficient video search systems. In this work we are proposing a framework for the video annotation enhancement and refinement with a flexible nature, that will enable this framework to accommodate and refine the annotation of any video corpus like YouTube, Video.com, TRECVID or any other multimedia corpus that have annotation in textual format.

5.3 State-of-the-Art

As a basic technique in video index and search, semantic-level video annotation (i.e., the semantic video concept detection) has been an important research topic in the multimedia research community [Naphade 2002; Snoek et al. 2006]. It aims at annotating videos with a set of concepts of interest, including scenes (e.g., urban, sky, mountain), objects (e.g., airplane, car, face), events (e.g., explosion-fire, people-marching) and certain named entities (e.g., person, place) [Naphade et al. 2005; Snoek et al. 2006]. Many efforts have been made on developing concept detection methods that can bridge the well-known semantic *"gap"* between the low-level features and high-level semantic concepts [Hauptmann et al. 2007]. Among these efforts, some have paid their attentions on detecting specific concepts, such as object detection based on the bag-of-feature model [Jiang et al. 2007]. Recently, more exploits to have been made on annotating video concepts in a generic style. For example, [Naphade et al. 2006] build large-scale concept ontology for generic video annotation and [Snoek et al. 2006] construct an ontology of 101 concepts from News video as well. In order to annotate these generic video concepts, [Yanagawa et al. 2007] build a set of baseline detectors for 374 LSCOM concepts [Naphade et al. 2006] by using Support Vector Machine (SVM) and [Wang et al. 2007] attempt to leverage diverse features to detect different video concepts. On the other hand, [Snoek et al. 2006] propose a novel pathfinder to utilize the authoring information to help index the generic multimedia data.

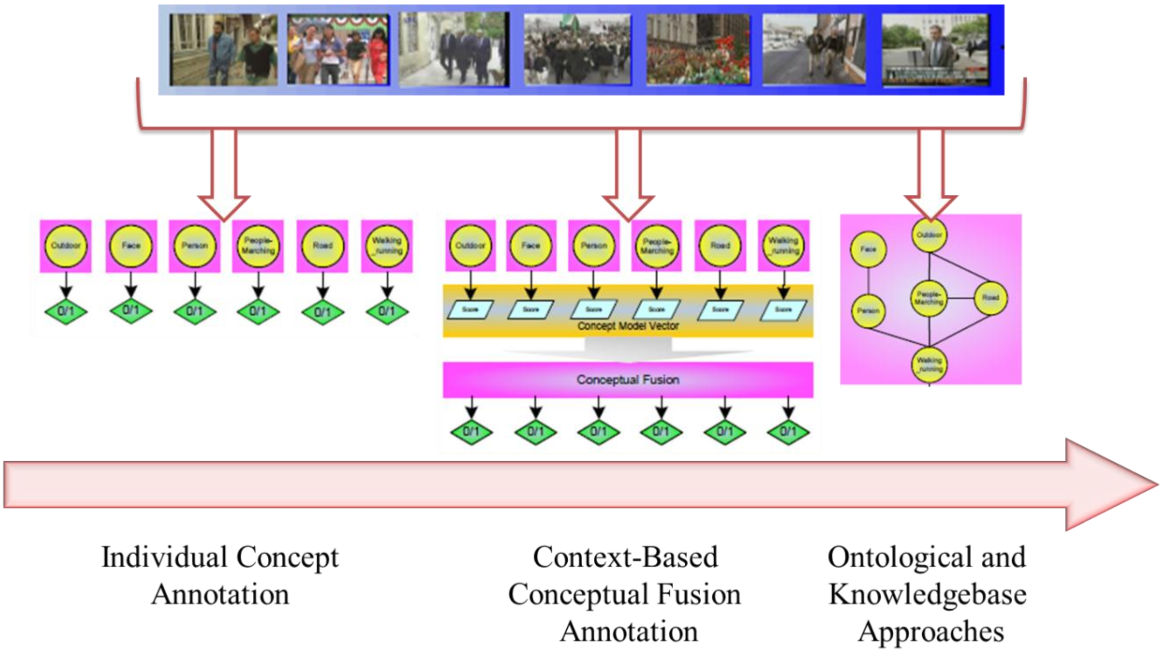


Figure 5.3:Video annotation model

In contrast to the above generic video annotation algorithms, multi-label video annotation process is another way, where a video can be annotated by multiple labels at the same time. These multi-labeled videos commonly exist in many real-world video corpuses, for example, most of the videos in the widely-used TRECVID dataset [Smeaton et al. 2006] are annotated by more than one label from a set of 39 different concepts. For example, a video can be classified as *"person," "walking running,"* and *"road"* simultaneously. In contrast to the multi-label problem, multiclass annotation only assigns one concept to each video. In most real-world video annotations, such as TRECVID annotations and the users' tags on many video-sharing website, the videos are often multi-labeled by a set of the concepts rather than only a single one. Next, we discussed the video annotation and divided the video annotation into three models.

5.3.1 Individual Concept Annotation

The annotation methods under this category are individual concept detectors; that is, they annotate the video concepts individually and independently as shown in the Figure 5.3. They ignore the rich relationships between the video concepts. In more detail, these methods translate the multi-label annotations into some independent concept detectors that individually assign presence/absence labels into each sample. Most classical detectors can be categorized into this model. For example, SVM [Cristianini, et al. 2000] with one-against-the-other strategy attempts to learn a set of detectors, each of which independently models the presence/absence of a certain concept. Other examples of this model include Maximum Entropy Models (MEM) [Nigam et al. 1999], Manifold Ranking (MR) [Tang et al. 2007] etc. As described, a set of unique SVMs is learned for video concept annotation independently. In brief, the core of this paradigm is to formulate the video annotation as a collection of independent binary classifiers.

However, in various real-world problems, video concepts do often subsist correlatively with each other, rather than appearing in isolation. So the individual annotation only achieves limited success. For example, the presence of *“Boat Ship”* often occurs together with the presence of *“water,”* while *“Boat Ship”* and *“Car”* commonly do not co-occur. On the other hand, compared to simple concepts which can be directly modeled from low-level features, some complex concepts, for example, *“People-Marching”* are really difficult to be individually modeled due to the semantic gap between these concepts and low-level features. Instead, these difficult concepts can be best inferred based on the label correlations with the other concepts. For instance, the existence of *“People-Marching”* can be improved if both *“Crowd”* and *“Walking Running”* occur in a video. Therefore, it will be very useful to exploit the label correlations when annotating the multiple concepts together.

5.3.2 Context-Based Conceptual Fusion Annotation

As a step towards more advanced video annotation, the second model is built atop the individual concept detectors. It attempts to refine the detection results of the binary concept detectors with a Context Based Concept Fusion strategy. Many algorithms can be categorized into this model. For example, [Wu et al. 2004] use an ontology-based multi-classification learning for video concept detection. Each concept is first independently modeled by a classifier, and then a predefined ontology hierarchy is investigated to improve the detection accuracy of the individual classifiers. Smith and [Naphade, et al 2003] present a two-step Discriminative Model Fusion approach to mine the unknown or indirect relationship between specific concepts by constructing model vectors based on detection scores of individual classifiers. A SVM is then trained to improve the detection outcomes of the individual classifiers. Alternative fusion strategy can also be used; for example, [Hauptmann et al. 2004] propose to use Logistic Regression to fuse the individual detections. Jiang et al. [2006] use a Context Based Concept Fusion-based learning method. Users are involved in their approach to annotate a few concepts for extra videos, and these manual annotations were then utilized to help infer and improve detections of other concepts. [Naphade et al. 2002] propose a probabilistic Bayesian Multi-Net approach to explicitly model the relationship between the multiple concepts through a factor graph which is built upon the underlying video ontology semantics. [Yan et al. 2006] mine the relationship between the detection results of different concepts by a set of various probabilistic graphical models. [Zha et al. 2007] propose to leverage the pairwise concurrent relations between different labels to refine the video detection output by individual classifiers of the concepts.

5.3.3 Ontological and Knowledgebase Approaches

The term *"Ontology"* refers to the theoretical representation model in knowledge systems [Hauptmann et al. 2007]. Some approaches tried to use Ontology to detect visual concepts. For example, in [Hauptmann et al. 2007], Ontology was built by learning concepts relationships based on analyzing co-occurrences between concepts. Other approaches have directly included visual knowledge in multimedia domain-specific Ontology, in a form of low-level visual descriptors for concept instances, to perform semantic annotation [Bagdanov et al. 2007]. As these methods almost depend on rules that are created by domain experts, they are subject to some inconsistency inherited from variations of the involved human culture, mood, personality, as well as the specific topic. In addition to that, they become almost less efficient in wider domains.

Research in text mining area conducts to build sizable commonsense knowledgebases. The Commonsense is the information and facts that are expected to be commonly known by ordinary people. Although, it may be considered as part of Ontology, we separate them to clarify the difference between domain-specific knowledge and commonsense knowledge.

In semantic video applications area, commonsense knowledgebases have recently received some attention to solve annotation issues, by finding related concepts. In [Yuan et al. 2008] concepts relationships are learned, in public video databases, using ConceptNet *"get_context"* functionality. WordNet [Felbaum. 1998] has been exploited in many applications in this area to find similar meaning annotations. For example, in [Shevade et al.

2006], a user, supported by WordNet, creates a visual concept for a group of images. Then ConceptNet is used to calculate the distance between the concepts. Most famous commonsense knowledgebases are WordNet [Felbaum. 1998], Cyc [Lenat et al. 1995] and ConceptNet [Liu, et al. 2004]. Currently, ConceptNet is considered to be the biggest commonsense database built from freely entered text. This knowledgebase is very rich in relationships, the number of assertions and the types of relationships.

Other approaches have directly included in the ontology an explicit representation of the visual knowledge, to perform reasoning not only at the schema level but also at the data level. [Bloehdorn, et al. 2005], defined a Visual Descriptors ontology, a Multimedia Structure ontology and a Domain ontology to perform video content annotation at semantic level. The Visual Descriptors ontology included concept instances represented with MPEG-7 visual descriptors. [Dasiopoulou, et al. 2005] have included in the ontology instances of visual objects. They have used as descriptors qualitative attributes of perceptual properties like color homogeneity, low-level perceptual features like components distribution, and spatial relations. Semantic concepts have been derived from color clustering and reasoning. [Maillot, et al. 2008] have proposed a visual concept ontology that includes texture, color and spatial concepts and relations for object categorization. A set of classifiers for the recognition of visual concepts is trained using features extracted from a set of manually annotated and segmented samples.

In the attempt of having richer annotations, other authors have explored the usage of reasoning over multimedia ontologies. In this case spatio-temporal relationships between concept occurrences are analyzed so as to distinguish between scenes and events and provide more precise and comprehensive descriptions. [Neumann, et al. 2006] have proposed a framework for scene interpretation using Description Logic reasoning techniques over *"aggregates",* these are composed of multiple parts and constrained by temporal and spatial relations to represent high-level concepts, such as objects conjurations, events and episodes. In [Espinosa, et al. 2007] manually annotated regions of images are used as visual representations of concepts, and relations between concept instances are obtained automatically. Inference from observation to explanation (abduction) is then used to check, among detected entities, relations and constraints that lead to consistent interpretation of image content. [Leslie, et al. 2007] have employed a two-level ontology of artistic concepts that includes visual concepts such as color and brushwork in the first level, and artist name, painting style and art period for the high-level concepts of the second level. A transductive inference framework has been used to annotate and disambiguate high-level concepts. In [Dasiopoulou, et al. 2008] automatically segmented image regions are modeled through low-level visual descriptors and associated to semantic concepts using manually labeled regions as training set. Context information is exploited to reduce annotation ambiguities. The labeled images are transformed into a constraint satisfaction problem (CSP) that can be solved using constraint reasoning techniques.

Several authors have exploited the ontology schema using rule-based reasoning over objects and events. [Snoek, et al. 2005] performed annotation of sport highlights using rules that exploited face detection results, superimposed captions, teletext and excited speech recognition, and Allen's logic to model temporal relations between the concepts in the ontology. [Francois, et al. 2005] defined a special formal language to define ontologies of events and used Allen's logic to model the relations between the temporal intervals of elementary concepts, so as to be able to assess complex events in video surveillance. [Hollink, et al. 2005] defined a set of rules in SWRL (Semantic Web Rule Language) to perform semi-automatic annotation of images of pancreatic cells. [Bai, et al. 2007] defined a soccer ontology and applied temporal reasoning with temporal description logic to perform event annotation in soccer videos. All these methods have defined rules that are created by human experts; thus, these approaches are not practical for the definition of a large set of rules.

[Benitez, et al. 2002] and [Benitez, 2005] took this idea further and suggested media ontology (MediaNet) to help to discover, summarize, and measure knowledge from annotated images in the form of image clusters, word senses, and relationships among them. MediaNet, a Bayesian network-based multimedia knowledge representation framework, is composed by a network of concepts, their relations, and media exemplifying concepts and relationships. The MediaNet integrates classifiers in order to discover statistical relationships among concepts. WordNet is used to process image annotations by stripping out unnecessary information. The summarization process implements a series of strategies to improve the images' description qualities, for example using WordNet and image clusters to disambiguate annotation terms (images in the same clusters tend to have similar textual descriptions).

[Benitez, 2005] also proposes a set of measures to evaluate the knowledge consistency, completeness, and conciseness. [Tansley, 2000] used a network at the concept level, and [Benitez, 2005] used the MediaNet network to capture the relations at both concept and feature levels. In addition, [Benitez, 2005] utilized WordNet, which captures human knowledge that is not entirely present in multimedia data.

5.4 Proposed Framework

We proposed a forth paradigm for video annotation, which is the extension of our previous work as discussed in chapter 03 for images. From the previous work it is noticed that semantic annotation in wide videos domain has two main issues: the first is pictorial features processing to gain knowledge about the contents, and the second is expressing this knowledge in annotation format which needs text processing. That was the inspiration for building a framework, which is the extended version of the previous work "framework for the annotation expansion and refinement using knowledgebases" as depicted in Figure 3.1 that helps in this paradigm.

The input to this framework is the textual annotated portion of the LabelMe videos, while output is the expanded form of the annotation lexically and commonsensically using the knowledgebases to increase the semantic space of the video annotated data corpus. The structure of the output is in LabelMe XML schema that makes them portable and usable for any search engine. The flexible nature of this framework makes them feasible and applicable to any video corpus. We have applied our research work on the LabelMe videos, the structure of the LabelMe video datasets structure is similar as that of the LabelMe images, as the video is the sequential combination of the images. Based on this, the LabelMe video is handled, and the other difference is that they are not only dealing the objects tracking, but also capture events in the videos. The user begins the annotation process by clicking control points along the boundary of an object to form a polygon. When the polygon is closed, the user is prompted for the name of the object and information about its motion. The user may indicate whether the object is static or moving and describe the action it is performing, if any. The user can further navigate across the video using the video controls to inspect and edit the polygons propagated across the different frames.

To correctly annotate moving objects, The LabelMe web tool allows the user to edit key frames in the sequence. Specifically, the tool allows selection, translation, resizing, and editing of polygons at any frame to adjust the annotation based on the new location and form of the object. For the event annotation, the users have an option to insert the event description in the form of sentence description. When the user finishes outlining an object, the web client software propagates the location of the polygon across the video by taking into account the camera parameters. Therefore, if the object is static, the annotation will move together with the camera and not require further correction from the user. With this setup, even with failures in the camera tracking, the user can correct the annotation of the polygon and continue annotating without generating uncorrectable artifacts in the video or in the final annotation.

5.5 Evaluation and Experimental Setup

The almost all of the annotation experiments focus on evaluating the system effectiveness. The effectiveness of the proposed system was investigated by using the same measure that we used for the images like concept diversity, enrichment ration and retrieval degree. The experiments were performed on LabelMe Videos. An overview of the LabelMe Videos is discussed in the next section.

5.5.1 LabelMe Videos Datasets

The LabelMe Videos are aim to create an open database of videos where users can upload, annotate, and download content efficiently. Some desired features include speed, responsiveness, and intuitiveness. They designed an easily accessible, open, and scalable annotation system to allow online users to label a database of real-world videos. Using the LabelMe labeling tool, they created a video database that is diverse in samples and accurate, with human guided annotations. They enriched their annotations by propagating depth information from a static and densely annotated image database. The basic intention of this annotation tool and database is that it can greatly benefit the computer vision community by contributing to the creation of ground truth benchmarks for a variety of video processing algorithms, as a means to explore information of moving objects.

They intend to grow the video annotation database with contributions from Internet users. As an initial contribution, they have provided and annotated a first set of videos. These videos were captured at a diverse set of geographical locations, which includes both indoor and outdoor scenes. Currently, the database contains a total of 1903 annotations, 238 object classes, and 70 action classes.

The most frequently annotated static objects in the video database are buildings (13%), windows (6%), and doors (6%). In the case of moving objects the order is persons (33%), cars (17%), and hands (7%). The most common actions are moving forward (31%), walking (8%), and swimming (3%).

5.5.2 Concept Diversity

We achieve a good improvement in term concept diversity for videos as well by adding the expanded terms from the lexically and commonsensically knowledgebases (see section 3.3.2). It has been raised in a noticeable degree also from 233 to 539 for LabelMe videos. This diversity achieves 131.33% in the topic indexed for LabelMe video corpus. The Figure 5.4 shows the comparison of concept diversity of the initial tags and the expanded tags for the LabelMe video dataset, where for LabelMe videos the initials tagged terms were passes through the process of purification (see section 3.3.1). The selected terms are further expanded lexically and commonsensically to produce more semantic space for the videos.

The Figure 5.4 demonstrates this increasing of all differentiated tags. It demonstrates that there are rich concepts exist in the LabelMe video corpus, where the purified selected terms were further extended using the lexical and commonsensical knowledgebase. We achieve a considerable improvement in terms of concept diversity for LabelMe videos. It is due to the fact, that the initially the videos dataset annotated with a baselines and is limited and that does not capture all the possible semantic interpretation of the videos. The initial annotation of the LabelMe videos consists of many unusual and noisy terms, prior to the expansion these noisy terms are needed to be prune. Before performing the expansion, we extract all those terms that contribute to the semantic description of the video. After the extraction of the terms (see data filtration process section 3.3.1), the expansion phase (see section 3.3.2) is performed that expand every single concept tagged with the videos lexically and conceptually. All these expansions contribute to such a huge increase in terms of concept diversity.

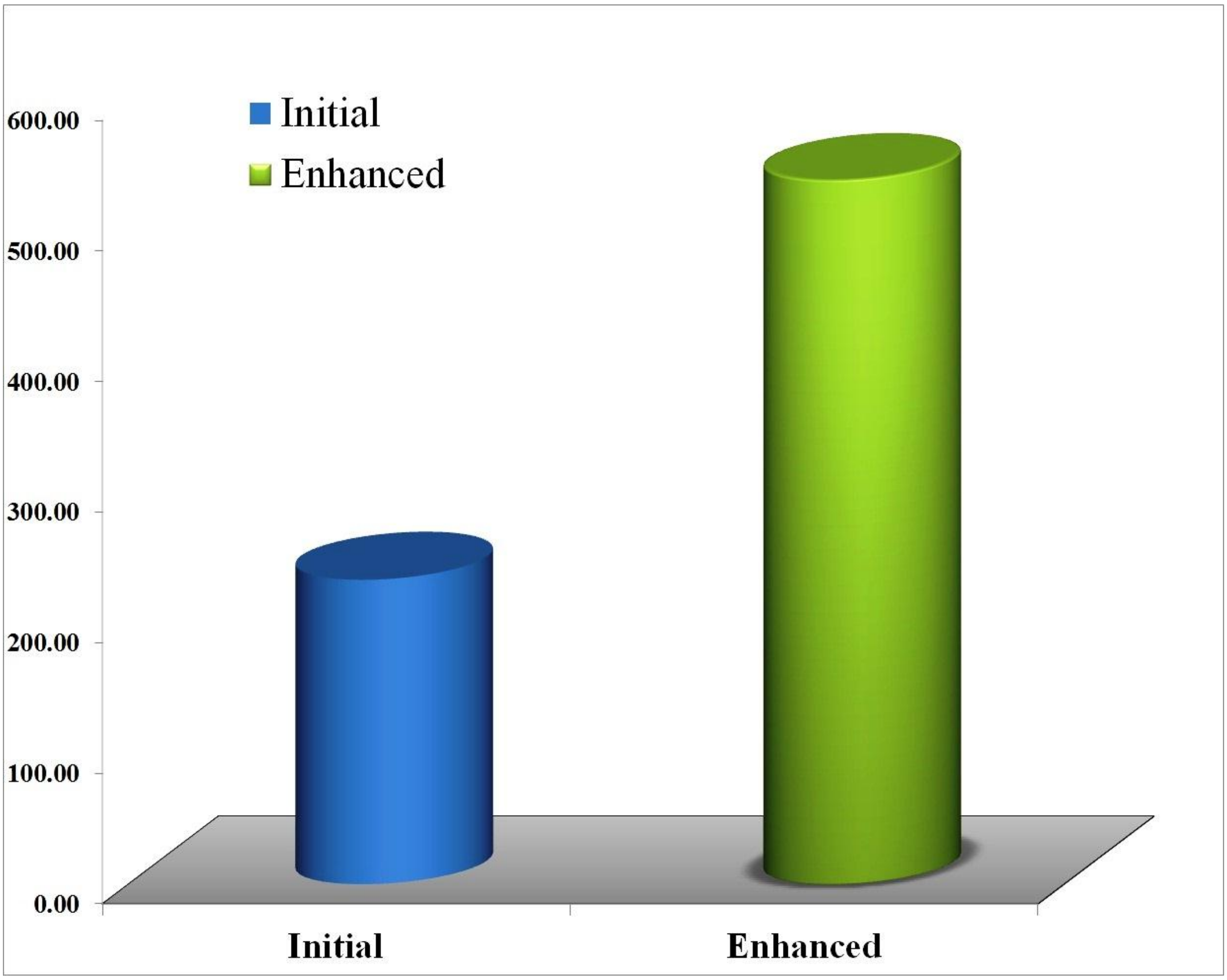


Figure 5.4: Shows the comparison of LabelMe video corpus in terms of Concept Diversity achieves before/after the process of the proposed framework.

This increase shows that the text mining approaches and usage of knowledgebases can benefit the annotation process and increase the semantic space of the multimedia which further helps in multimedia content understanding on one side while achieve a highly retrieval accuracy on the other side and can perform the worst queries with good results.

5.5.3 Enrichment Ratio

The tagging ratio (see section 3.4.2) for LabelMe video has been rise from 14.53 tags per video to 19.78 tags after enhancement and refinement (see section 3.3.2), whilst enrichment ratio has achieved a considerable degree about 136.13%. The Figure 5.5 shows the tagging ratio for the 10 sample randomly selected videos from the LabelMe videos dataset.

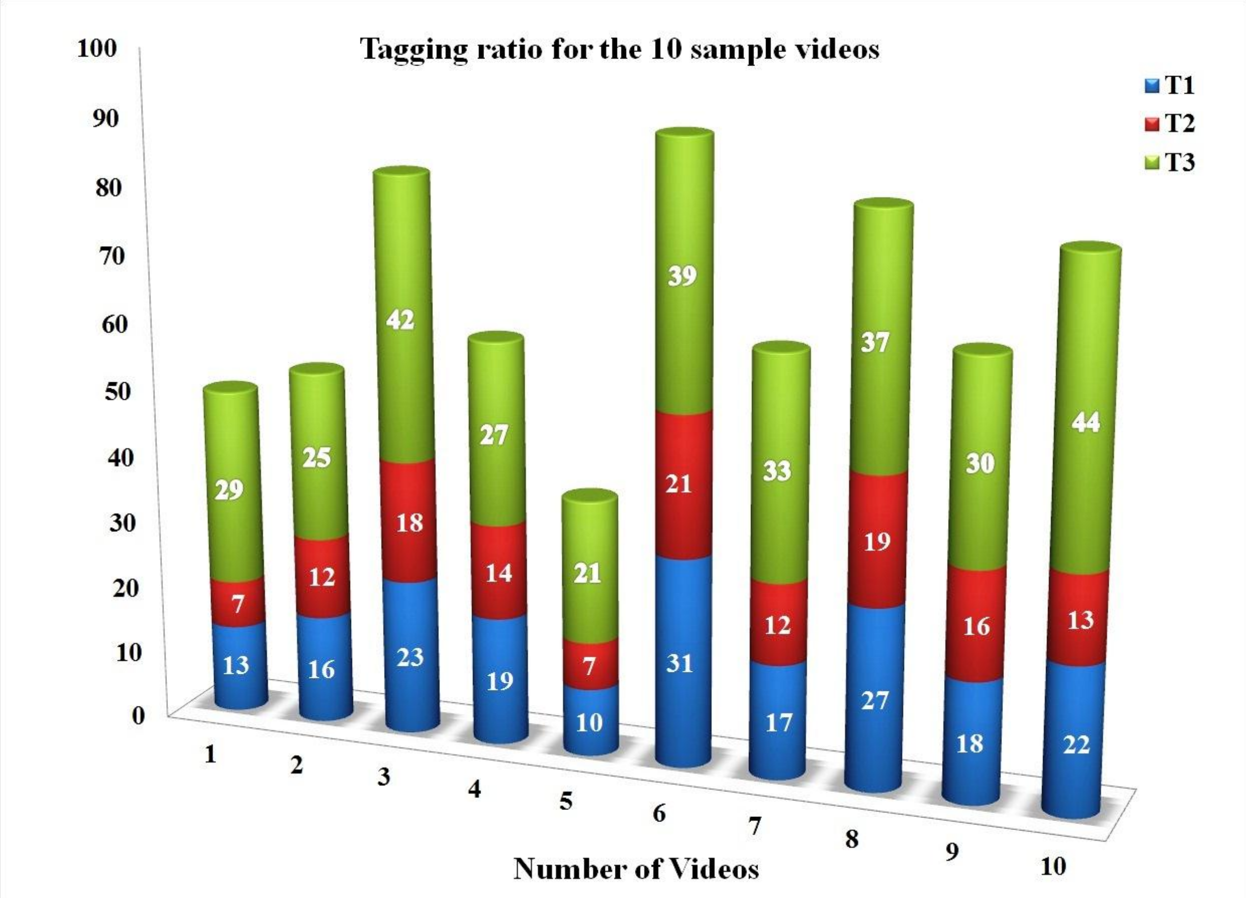


Figure 5.5: shows the number of tags per video of the 10 sample randomly selected videos taken from the LabelMe video dataset, where T_1 and T_2 represents the number tags before and after data filtration process, while T_3 shows number of tags after the annotation enhancement and refinement phase.

The Figure 5.5 depicts the tagging ratio of the randomly selected 10 sample videos. Originally, the videos were tagged with the terms, where some of the terms were unusual and noisy which is delineated by T_1In the proposed framework, the initial tag terms were first needed to be prune from these noisy terms (see section 3.3.1) and then the selected terms are passed to the next phase of the proposed framework i.e. the expansion phase (see section 3.3.2). The output of the initial refinement is represented by T_2. The refine tagged terms are then passed to the expansion phase to cover all the possible semantics dimensions of the images. The outcome increased in the tags per image of the expansion phase is delineated by T_3, which is the ratio between the refine and expanded lexical and conceptual terms. For instance, the image V_1 in Figure 5.5 is initially tagged with $T_1 \leftarrow 13$, these tags are then refined to $T_2 \leftarrow 7$. This decreases the number of tags as there were six unusual terms removed in the filtration process and filter out only those terms which contribute to the actual meaning behind the group of an object that constitutes the videos. After the expansion, the number of tags per image became $T_3 \leftarrow 29$, which raised the tagging ratio 314.29%. Similarly the increase in the tag for $V_2 \leftarrow 108.33\%, V_3 \leftarrow 133.33\%$, $V_4 \leftarrow 92.86\%$, $V_5 \leftarrow 200\%$, $V_6 \leftarrow 85.71.5\%$, $V_7 \leftarrow 175\%$, $V_8 \leftarrow 94.74\%$, $V_9 \leftarrow 87.5\%\ and\ V_{10} \leftarrow 238.46\%$ respectively. The rate of an increase in the tagging ratio for the 10 sample videos is different. It is because some of the videos are simple while some of them are semantically enriched. The concepts in the simple videos are limited so their semantic space will be small and therefore, their expansion will be limited. While for the semantically enriched videos consist of a large number of concepts and constitute a large semantic space as a result, the percentage increase in the tagging ratio will be large, because, the expansion is applied on every single term of the filter out terms lexically and commonsensically.

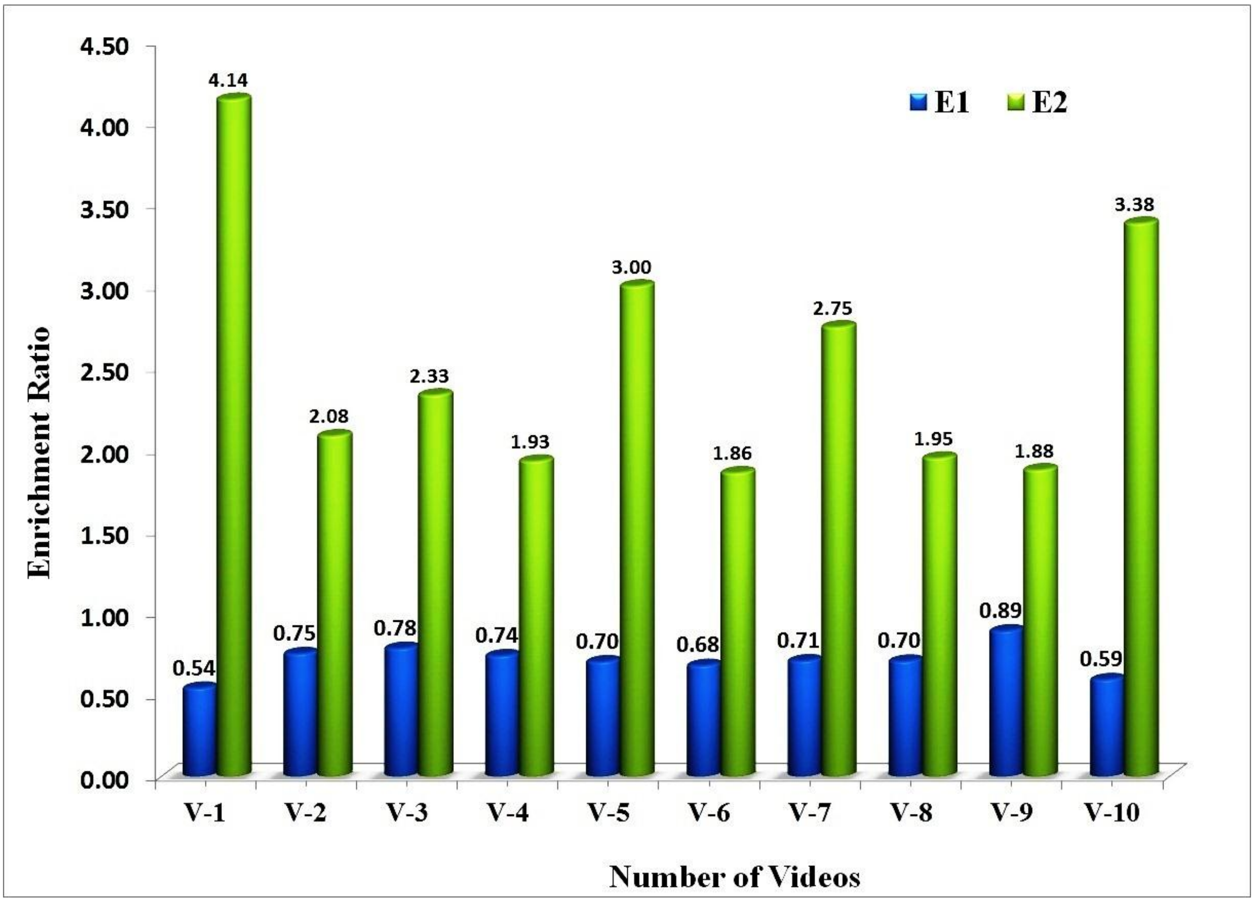


Figure 5.6: Graph shows the Enrichment ration between the $\boldsymbol{E_1}$ and $\boldsymbol{E_2}$ before/after the processing of the proposed framework

The Figure 5.6 shows the enrichment ratio for the same randomly selected 10 sample videos. The large gap among $\boldsymbol{E_1}$ and $\boldsymbol{E_2}$ are due to the fact, as the tags $\boldsymbol{T_1}$ are the baseline tags with the videos, while $\boldsymbol{T_2}$ are the filter out representation of the same tags which is for the most videos are same or less, so the enrichment ratio for this, .i.e. $\boldsymbol{E_1}$ will always be equal or less than 1. While for $\boldsymbol{E_2}$, the ratio is based on the $\boldsymbol{T_2}$ and $\boldsymbol{T_3}$, where $\boldsymbol{T_3}$ is representing the expanded tags which is for most of the images is greater than $\boldsymbol{T_2}$. So the enrichment ratio $\boldsymbol{E_2}$ will always be greater than or equal to 1. In the Figure 5.6, for example $\boldsymbol{V_1}$ have the highest $\boldsymbol{E_2}$ value among the others, which is due to the fact that the terms tag with the image $\boldsymbol{V_1}$ has a large number of lexical and conceptual expansion while the $\boldsymbol{V_6}$ have smaller $\boldsymbol{E_2}$ value is not only due to the small number of lexical and conceptual expansion but also the expanded terms are repeated, which were removed in the concept refinement phase (see section 3.3.4).

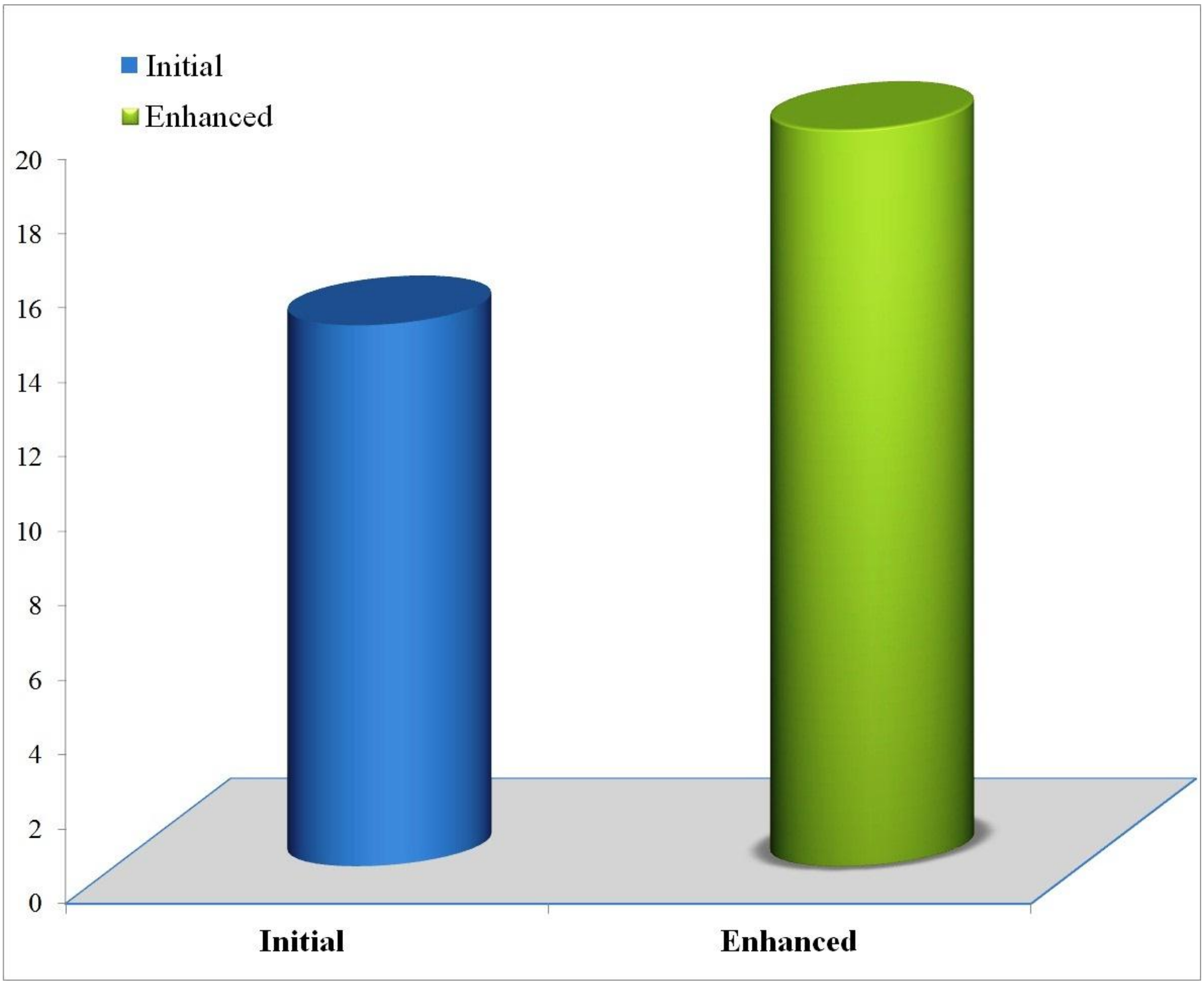


Figure 5.7: Shows the Enrichment ratio for the LabelMe video dataset.

The Figure 5.7 depicts the overall enrichment ratio for the LabelMe videos datasets. A substantial improvement has been achieved in term of enrichment ratio. The initially graph represents the enrichment of the tags before processing of the proposed framework, while the enhanced graph represents the enrichment ratio achieves after performing processing on the video corpus by using the proposed framework. The enrichment ratio has achieved a considerable degree about 136.13%. The fact behind this is that the actual annotation of the LabelMe video of the limited number of concepts tagged with the each of the video. Much of the relevant worthwhile information is available in the corpus, but fails to retrieve due to the different words used in the tagging, even though they contain same semantic ideas. We attempt to remove this bottleneck of the baseline approach by using the expansion techniques. The proposed technique select some of the most related expanded terms by computing the semantic similarity among the terms during the concept refinement phase (see section 3.3.4). This increases the enrichment ratio of the annotation and contributes in the semantic space enhancement of the videos. The higher the enrichment ratio, the higher is the semantic space for the videos and as a result increases the precision of the query even for a worst query as well.

5.5.4 Retrieval Degree

The evaluation of the proposed framework in terms of retrieval degree is to validate the performance of the proposed techniques. The retrieval degree is the number of relevant video retrieved as a result of a query applied on the corpus and as a result depicts the annotation efficiency of the proposed techniques. We perform the experiments by using the same query engine that we have used for the images, i.e. LabelMe query engine, which work on the principle of string matching techniques for search and retrieval. Using the proposed framework, the retrieval degree has been increased. We investigate the retrieval degree of the proposed framework in terms of precision and recall. The main focus of our research is to bridge the semantic gap by achieving the precise and accurate results. As we know, that the expansion sometimes leads to too many results that will increase the recall but significantly decrease the precision of the system. The decrease in precision is due to the fact that among the expanded terms some of them are most relevant than the others. We have maintained the precision of our proposed system by selecting some of the most relevant terms by using the refinement module.

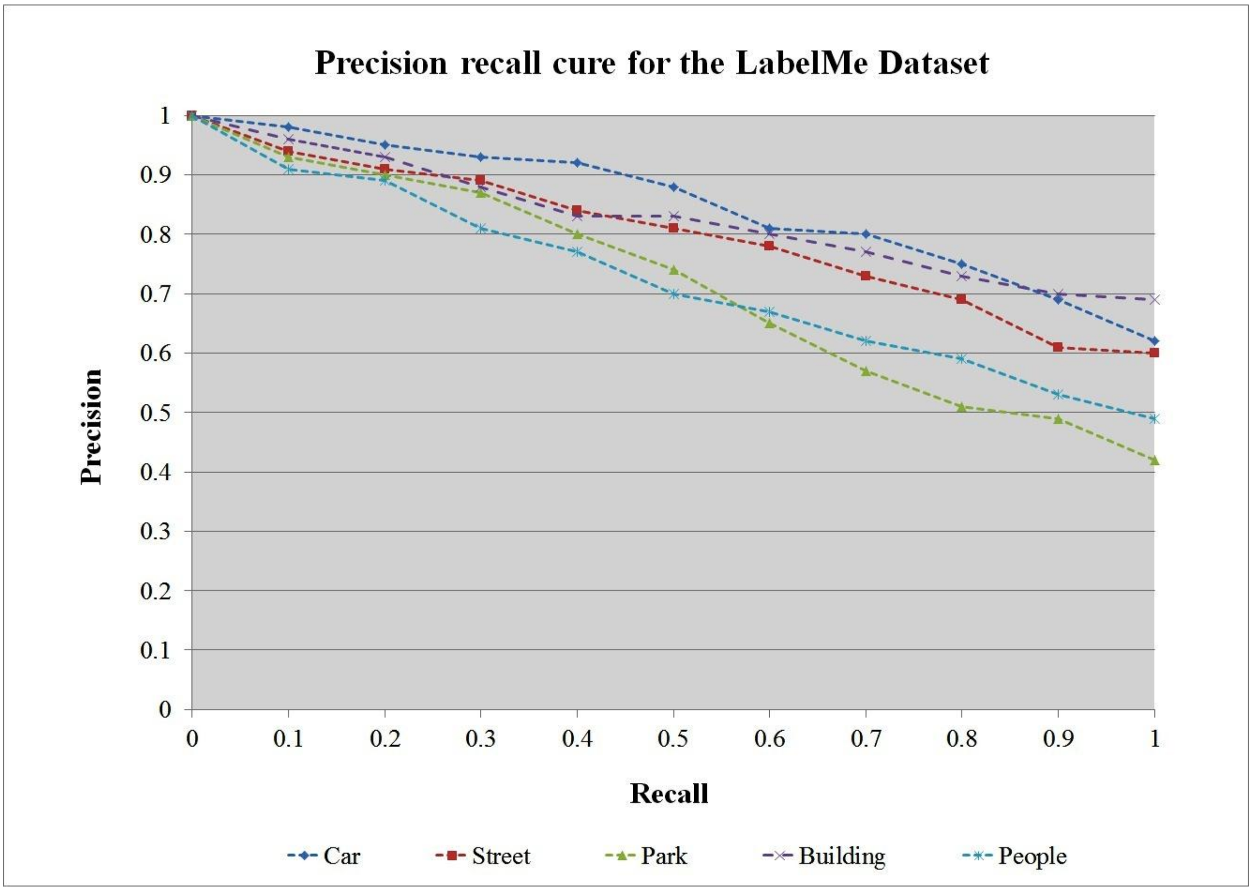


Figure 5.8: Precision recall curve for the top 10 queries result on the LabelMe video corpus.

The Figure 5.8 shows the precision recall cure for the top 10 results of the five randomly selected concepts over the LabelMe video's dataset. The Figure 5.8 depicts the significant outcome in terms of precision of the proposed framework. The randomly selected concepts may belong from any category of the concept, i.e. single word single concept or single word multi-concept. Among the concepts, the *car* is the simple single word single concept and is easy to deal with. Most of the traditional systems are able to handle such a type of systems but flunks to deal with the complex concepts. The mean average precision of the randomly selected concepts is as for the concept *Car* the mean average precision is 0.85, for concept *Street* it is 0.80, for *Park* 0.72, while for *Building* it is 0.83 and for *People* mean average precision is 0.73 respectively. The variation in the outcome of the various concepts is due to the nature of the concept, as with the increase in the complexity of the concept, there will be the decrease in the performance and accuracy of the system. The concept like *building* which are the single word multi-concept or abstract concepts, because it contains further other concepts like a *home*, *apartment*, *house*, and *shop*, etc., i.e. they are different conceptual terms for the word *building*. While for the concepts like *park*, the outcome of the proposed framework is not significant. It is due to the fact that concepts like *park*, *jaguar*, *apple*, etc., are the ambiguous concepts. Humans can easily recognise the difference in the multimedia (moving or still images) of the *car park* and the *recreational park* while a computer can't. All this is due to the difference in the flexible human nature and hard coded form of computer nature. The complexity of such a type of concepts can be reduced by the length of the query, because these words help to identify the category of *park*. In our investigation of the proposed framework, we have used only the single word concept that's why the performance of the proposed framework over the concept *park* is not significant. While the other concepts like *street* and *people* in the Figure 5.8 are the multi-concepts but only have a single interpretation and are not ambiguous like *park*.

5.6 Chapter Summary

In this chapter, we have presented a semantic enhancement and refinement approach for the videos. We have investigated the semantic enhancement and refinement on LabelMe video dataset. The proposed technique shows substantial results for the LabelMe videos. We have used the concept diversity, enrichment ratio and retrieval degree based on the precision and recall to test the efficiency of the proposed semantic query interpreter on the video datasets. Experimental results for the LabelMe video data set have demonstrated the usefulness of the proposed semantic based extraction.

Chapter 06

Conclusion & Perspectives

"Solutions almost always come from the direction you least expect, which means there's no point in trying to look in that direction because it won't be coming from there."

The Salmon of Doubt by Douglas Noel Adams

The basic purpose behind this chapter is giving a final reflection on the finished work and explores the directions for future work. We have addressed the main challenge of Semantic gap in Semantic Multimedia analysis and annotation. We have tried to reduce this gap. This dissertation has proposed solutions to the problems that help in the extraction and exploitation of the actual semantics inside the image and the video using the open source knowledge bases.

This chapter draws a conclusion in summarizing its cognitions and illustrates the course of the work. Section 6.1 summaries the findings of this thesis. In Section 6.2 , the works that have not been considered in this research but that are worth being focused on in a future work.

6.1 Research Summary

Aiming to bridge the semantic gap, this thesis is presented a new paradigm of semantic based video and image search, more specifically, concept based video and image search method where the knowledge bases are used to extract the semantics in order to find the users requirements.

The following contributions have been presented in this thesis:

6.2.1 A Framework for Images Annotation Enhancement & Refining Using Knowledge Bases

This first contribution of this dissertation is to propose a Framework for Images Annotation Enhancement & Refining using Knowledgebases. The role of the knowledgebase for high level semantic annotation has been recognized in the literature. Based on this, we used the open source knowledgebases (i.e. WordNet and ConceptNet) as a first step towards high level semantic annotation, where already object/concept based annotated corpora are passed through the process of the proposed framework. We have selected LabelMe images datasets for the said purpose, which is created by using the web tool where a user has a free hand to sketch the edges of the object in the image and tag with the user define keyword/concept, as a result problem like redundancy, irrelevant and unusual keywords/concepts tag with the objects are continuously generated. So the emphasis of the proposed work is to first purify the dataset by using the redundancy control, unification, stopwords algorithms. The WordNet and ConceptNet are utilized to expand the concepts lexically and commonsensically, the reason for using such knowledgebases is two ford, (1) both of them are open sources and is freely available for research (2) they have natural language form with semantic relational structure. Adding to this, the ConceptNet nodes mainly address everyday life and have the ability to connect concepts and their events and hence suitable for commonsensical expansion, while WordNet nodes mainly on formal taxonomies and support the single words and having a support for synsets which is useful for lexical expansion of the said corpus. The lexical and commonsensical expanded form comes up with too many keywords. Some of them are irrelevant and noisy that decreases the precision of the query. For the better precision, we have to remove these noisy keywords. For refinement, we applied semantic similarity among the original and each of the generated keywords and discard the keywords that fails to achieve the defined threshold. The result of the experiments exposes that the proposed framework achieve the substantial improvement in terms of concept diversity, enrichment ratio and retrieval degree. The proposed system has been implemented by using Matlab and C# environment. The source code of the proposed contribution is available in Appendix.

6.2.2 High Level Semantic Propagation

The proposed framework discussed in section 6.2.1, solve the lexical and vocabulary gap for the concept based annotation techniques, but feebly answer to the problem of high level semantic annotation. As it is commonly understood that the progression in automatic annotation have not been able to comprehend with adequately accurate results, to outfit multimedia (e.g. image/video) retrieval capabilities, digital libraries have hung on manual annotation of images. Providing a track to enact high level semantic annotation automatically would be more worthwhile, efficient and scalable with magnifying image collections. This contribution intent to equip the high level semantic annotation for images by calculation first the semantic intensity (SI) of the concept in the image which is the dominancy factor of the concept, as we are aware of the fact that the image is the combination of various concepts and among the list of concepts some of them are more dominant then the other. Secondly the semantic similarities of the images are calculated on the basis of concept similarity and their SI values tag with the image. To ease the process HLS propagation process, a clustering technique for each of the image are applied, where a set of full similar (FS) and partial similar (PS) are prepared on the basis of image similarity. The images having similarity values greater than or equal to 0.80 are cluster under FS set, while having value greater than or equal to 0.50 are a part of PS set. This approach facilitate the annotator in term of annotation accuracy, where a single effort of the human experts to assign high level semantic to a randomly selected image and propagate to other images through clustering for other images. The experiment on a portion of randomly selected images from LabelMe database manifests stimulating outcomes. The proposed system has been implemented using the Matlab and C# environment which is available in appendix.

6.2.3 Annotation Enhancement & Refinement for Video

The efflux of multimedia is not comes in images but for video as well. After investigating the effectiveness of the proposed framework for images annotation enhancement and refinement, have been extends to video domain to investigate its performance on video as well. We have exercised the similar approaches on the LabelMe video datasets. The LabelMe video annotation structure is similar is that of the images with extra information for every frame and handle the events as well. The temporal information is recorded per frame, where the changes in object location and size are control by the users. The process of lemmatization, stopwords, unification and redundancy control are performed. The purification processes are conducted on the dataset to purify them and then expand the concepts tag with the video lexically and commonsensically with the aid of WordNet and ConceptNet and then semantic similarity for further purify the concepts. The experimental results have been made in terms of concept diversity, enrichment ratio and retrieval degree to ensure the performance of the proposed work and a noticeable improvement has been achieved. The proposed system has been implemented using the Matlab and C# environment available in appendix.

6.2 Future Perspective

The problems addressed by this dissertation are very challenging. This dissertation aims at providing a solution to semantic modeling and interpretation for image and video annotation. We have tried to propose a system that better satisfy the users' demands and needs. Although encouraging performance has been obtained by using proposed contributions but some of the work are worth investigating and needs further extension. In this section, we discuss some of the remaining issues in our proposed solutions.

6.2.1 Integration of Cyc Knowledgebase to the Annotation Enhancement & Refinement Framework

The proposed annotation enhancement and refinement framework is worth to be extended by integrating the Cyc knowledgebase. The Cyc is the largest open source knowledgebase. The Cyc is not rich in conceptual reasoning like the ConceptNet and lexically rich like WordNet. But contain more information than ConceptNet and WordNet. Some of the terms that are missing in WordNet and ConceptNet are available in Cyc. The latest version of OpenCyc, 2.0, was released in July 2009. OpenCyc 1.0 includes the entire Cyc ontology containing hundreds of thousands of terms, along with millions of assertions relating the terms to each other, however, these are mainly taxonomic assertions, not the complex rules available in Cyc. The knowledge base contains 47,000 concepts and 306,000 facts and can be browsed on the OpenCyc website. This will make the proposed framework for annotation enhancement and refinement more flexible.

6.2.2 LabelNet: A Conceptual shape based knowledgebase of the LabelMe image and video dataset

LabelMe consists of the set of images which are annotated with the list of objects. These objects are represented by the set of polygon values. These polygons constitute the shape and the area of the objects. We will try to make worth of these polygons to constitute the shape based knowledgebase known as LabelNet. LabelNet attaches a concept to a particular shape in the LabelMe image dataset. These concepts are already expanded by the integration of three knowledgebases i.e. WordNet, ConceptNet and Cyc. The LabelNet tags all the possible shapes of the particular concepts. Analogy to the textual synonyms it will make the shape synonym. Let’s take a simple scenario of a particular concept car. The

LabelNet tag a concept car with all the possible shapes of the car available in the LabelMe dataset. The LabelNet makes the shape based ontology of the concepts available in the LabelMe. The basic intention of this model is to bridge the semantic gap by integrating the knowledgebases and the low level shape based retrieval. The LabelNet will also make the object detection.

6.2.3 Automatic Object Detection for the LabelMe

The main bottleneck of the LabelMe system is that it’s manual annotation framework, where the users manually annotate the objects that are represented by the set of polygons. We will try to develop an automated object detection system that will detect all the shapes available in the LabelMe images and video frames specifically and other images and video in general. These shape or the automatic object detection system are then integrated into the LabelNet to convert these primitive information into the semantic level.

6.2.4 Extension of High Level Semantic Propagation for LabelMe videos

The high level semantic propagation outperforms for the images and we will extend this to video domain as well. We will also investigate the performance of the proposed contributions on other image and video data like TRECVID, ImageCLEF, Corel, YouTube etc.

Appendix

1. Matlab Source Code

1.1. Setting Path to the annotation and image/video corpus

```
// This function set path to the LabelMe annotation source folder
function setAnnotationPath(Path)
global HA;
HA = Path;


// This function set path to the LabelMe images source folder
function setImagePath(Path)
global HI;
HI = Path;


// This function set path to the LabelMe images source folder
function setVideoPath(Path)
global HV;
HV = Path;
```

1.2. Database Creation

```
// This function create a virtual database for the experiments
function Report = DBCreation
global DB HA;
DB = LMdatabase(HA);
Report = 'Database creation completed';

// Source function for the creation of the database from the LabelMe XML
files

function [D, XML] = LMdatabase(varargin)
Folder = [];

% Parse input arguments and read list of folders
Narg = nargin;
HOMEANNOTATIONS = varargin{1};
if Narg==3
    HOMEIMAGES = varargin{2};
else
    HOMEIMAGES = '';
end

if iscell(varargin{Narg})
    if Narg == 2
        Folder = varargin{2};
        Nfolders = length(Folder);
```

```
    end
    if Narg == 3
        Folder = varargin{3};
        Nfolders = length(Folder);
    end
    if Narg == 4
        Folder = varargin{3};
        Images = varargin{4};
        Nfolders = length(Folder);
    end
else
    if Narg==2
        HOMEIMAGES = varargin{2};
    end
    if ~strcmp(HOMEANNOTATIONS(1:5), 'http:');
        folders = genpath(HOMEANNOTATIONS);
        h = [findstr(folders,  pathsep)];
        h = [0 h];
        Nfolders = length(h)-1;
        for i = 1:Nfolders
            tmp = folders(h(i)+1:h(i+1)-1);
            tmp = strrep(tmp, HOMEANNOTATIONS, ''); tmp = tmp(2:end);
            Folder{i} = tmp;
        end
    else
        files = urldir(HOMEANNOTATIONS);
        Folder = {files(2:end).name}; % the first item is the main path name
        Nfolders = length(Folder);
        %for i = 1:Nfolders
        %    Folder{i} = Folder{i};
        %end
    end
end

% Open figure that visualizes the file and folder counter
Hfig = plotbar;

% Loop on folders
D = []; n = 0; nPolygons = 0;
if nargout == 2; XML = ['<database>']; end
for f = 1:Nfolders
    folder = Folder{f};
    disp(sprintf('%d/%d, %s', f, Nfolders, folder))


    if Narg<4
        filesImages = [];
        if ~strcmp(HOMEANNOTATIONS(1:5), 'http:');
            filesAnnotations = dir(fullfile(HOMEANNOTATIONS, folder,
'*.xml'));
            if ~isempty(HOMEIMAGES)
                filesImages = dir(fullfile(HOMEIMAGES, folder, '*.jpg'));
            end
        else
            filesAnnotations = urlxmldir(fullfile(HOMEANNOTATIONS, folder));
            if ~isempty(HOMEIMAGES)
```

```
                filesImages = urldir(fullfile(HOMEIMAGES, folder), 'img');
            end
        end
    else
        filesAnnotations(1).name = strrep(Images{f}, '.jpg', '.xml');
        filesAnnotations(1).bytes = 1;
        filesImages(1).name =  strrep(Images{f}, '.xml', '.jpg');
    end

    %keyboard

    if ~isempty(HOMEIMAGES)
        N = length(filesImages);
    else
        N = length(filesAnnotations);
    end

    %fprintf(1, '%d ', N)
    emptyAnnotationFiles = 0;
    labeledImages = 0;
    for i = 1:N
        clear v
        if ~isempty(HOMEIMAGES)
            filename = fullfile(HOMEIMAGES, folder, filesImages(i).name);
            filenameanno = strrep(filesImages(i).name, '.jpg', '.xml');
            if ~isempty(filesAnnotations)
                J = strmatch(filenameanno, {filesAnnotations(:).name});
            else
                J = [];
            end
            if length(J)==1
                if filesAnnotations(J).bytes > 0
                    [v, xml] = loadXML(fullfile(HOMEANNOTATIONS, folder,
filenameanno));
                    labeledImages = labeledImages+1;
                else
                    %disp(sprintf('file %s is empty', filenameanno))
                    emptyAnnotationFiles = emptyAnnotationFiles+1;
                    v.annotation.folder = folder;
                    v.annotation.filename = filesImages(i).name;
                end
            else
                %disp(sprintf('image %s has no annotation', filename))
                v.annotation.folder = folder;
                v.annotation.filename = filesImages(i).name;
            end
        else
            filename = fullfile(HOMEANNOTATIONS, folder,
filesAnnotations(i).name);
            if filesAnnotations(i).bytes > 0
                [v, xml] = loadXML(filename);
                labeledImages = labeledImages+1;
           else
                disp(sprintf('file %s is empty', filename))
                v.annotation.folder = folder;
```

```
                v.annotation.filename = strrep(filesAnnotations(i).name,
'.xml', '.jpg');
            end
        end

        n = n+1;

        % Convert %20 to spaces from file names and folder names
        if isfield(v.annotation, 'folder')
            v.annotation.folder = strrep(v.annotation.folder, '%20', ' ');
            v.annotation.filename = strrep(v.annotation.filename, '%20', '
');

            % Add folder and file name to the scene description
            if ~isfield(v.annotation, 'scenedescription')
                v.annotation.scenedescription = [v.annotation.folder ' '
v.annotation.filename];
            end
        end


        % Add object ids
        if isfield(v.annotation, 'object')
            %keyboard
            Nobjects = length(v.annotation.object);
            [x,y,foo,t,key] = LMobjectpolygon(v.annotation);

            % remove some fields
            if isfield(v.annotation.object, 'verified')
                v.annotation.object = rmfield(v.annotation.object,
'verified');
            end

            for m = 1:Nobjects
                % lower case object name
                if isfield(v.annotation.object(m), 'name')
                    v.annotation.object(m).name =
strtrim(lower(v.annotation.object(m).name));
                end

                % add id
                if isfield(v.annotation.object(m).polygon, 'pt')
                    v.annotation.object(m).id = m;

                    % Compact polygons
                    v.annotation.object(m).polygon =
rmfield(v.annotation.object(m).polygon, 'pt');

                    pol.x = single(x{m});
                    pol.y = single(y{m});
                    pol.t = uint16(t{m});
                    pol.key = uint8(key{m});
                    if isfield(v.annotation.object(m).polygon, 'username')
                        pol.username =
v.annotation.object(m).polygon.username;
```

```
                    end
                    v.annotation.object(m).polygon = pol;
                else
                    v.annotation.object(m).deleted = '1';
                end
            end
        end

        % store annotation into the database
        D(n).annotation = v.annotation;

        if nargout == 2
            XML = [XML xml];
        end

        if mod(i,10)==1 && Narg<4
            plotbar(Hfig,f,Nfolders,i,N);
        end
    end
    disp(sprintf(' Total images:%d, annotation files:%d (with %d empty xml
files)', N, labeledImages, emptyAnnotationFiles))
end

if nargout == 2; XML = [XML '</database>']; end

% Remove all the deleted objects. Comment this line if you want to see all
% the deleted files.
D = LMvalidobjects(D);

% Add view point into the object name
D = addviewpoint(D);

% Add crop label:
%words = {'crop', 'occluded', 'part'};
%D = addcroplabel(D, words); % adds field <crop>1</crop> for cropped objects



disp(sprintf('LabelMe Database summary:\n Total of %d annotated images.',
length(D)))
%disp('-----------------')
%
close(Hfig)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%


// buildin function for progress bar
function fig = plotbar(fig,nf,Nf,ni,Ni)

if nargin > 0
    clf(fig)
    ha = subplot(2,1,1, 'parent', fig); cla(ha)
```

```
    p = patch([0 1 1 0],[0 0 1 1],'w','EraseMode','none', 'parent', ha);
    p = patch([0 1 1 0]*nf/Nf,[0 0 1
1],'g','EdgeColor','k','EraseMode','none', 'parent', ha);
    axis(ha,'off')
    title(sprintf('folders (%d/%d)',nf,Nf), 'parent', ha)
    ha = subplot(2,1,2, 'parent', fig); cla(ha)
    p = patch([0 1 1 0],[0 0 1 1],'w','EraseMode','none', 'parent', ha);
    p = patch([0 1 1 0]*ni/Ni,[0 0 1
1],'r','EdgeColor','k','EraseMode','none', 'parent', ha);
    axis(ha,'off')
    title(sprintf('files (%d/%d)',ni,Ni), 'parent', ha)
    drawnow
else
    % Create counter figure
    screenSize = get(0,'ScreenSize');
    pointsPerPixel = 72/get(0,'ScreenPixelsPerInch');
    width = 360 * pointsPerPixel;
    height = 2*75 * pointsPerPixel;
    pos = [screenSize(3)/2-width/2 screenSize(4)/2-height/2 width height];
    fig = figure('Units', 'points', ...
        'NumberTitle','off', ...
        'IntegerHandle','off', ...
        'MenuBar', 'none', ...
        'Visible','on',...
        'position', pos,...
        'BackingStore','off',...
        'DoubleBuffer','on');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
function files = urlxmldir(page)

files = []; Folder = [];
page = strrep(page, '\', '/');

%page

[folders,status] = urlread(page);
if status
    folders = folders(1:length(folders));
    j1 = findstr(lower(folders), '<a href="');
    j2 = findstr(lower(folders), '</a>');
    Nfolders = length(j1);

    fn = 0;
    for f = 1:Nfolders
        tmp = folders(j1(f)+9:j2(f)-1);
        fin = findstr(tmp, '"');
        if length(findstr(tmp(1:fin(end)-1), 'xml'))>0
            fn = fn+1;
            Folder{fn} = tmp(1:fin(end)-1);
        end
    end

    for f = 1:length(Folder)
```

```
        files(f).name = Folder{f};
        files(f).bytes = 1;
    end
end
```

1.3. Output Display

```
function resultDisplay
global Dq HI HA;

for n = 1: 5
    fn =
fullfile(HA,Dq(n).annotation.folder,strrep(Dq(n).annotation.filename,'.jpg','
.xml'));
    [annotation img] = LMread(fn, HI);
    objName = '';
    for i = 1: length(annotation.object)
        objName = strcat(objName,', ', annotation.object(i).name);
    end
    figure;
    imshow(img);
    title(objName);
end
```

1.4. Semantic Intensity Calculation

```
function SemanticIntensity(HI, HA, nHA)

% Reading XML files from the folders
dirList = dir(HA);

% Performing file wise operation
for n = 3:length(dirList)
   dirPath = strcat(HA,'\',dirList(n).name);
  fileList = filenames(dirPath,'xml',2);

  if (~strcmpi(fileList,'Irfan'))
      for i = 1:length(fileList)
          [annotation, img] = LMread(fullfile(dirPath, fileList{i}),HI);
          [h w] = size(img);
          NI = h * w;

          % Calculate Semantic Intensity of each object
          if isfield(annotation,'object')
              for j = 1:length(annotation.object)
                  [X,Y] = getLMpolygon(annotation.object(j).polygon);
                  SI = polyarea(X,Y)/NI;
                  annotation.object(j).name =
strcat(annotation.object(j).name,' (',num2str(SI),')');
                  annotation.object(j).SI = SI;
              end
```

```
            v.annotation = annotation;
            writeXML(fullfile(nHA, annotation.folder,fileList{i}),v);
        end
    end
  end
end
disp(' Irfan -- Semantic Intensity operation completed successfully');
```

1.5. Redendancy Control

```
function uSemanticIntensity(HI, HA, nHA)

% Reading XML files from the folders

% Load replacewords list
load('D:\Research\LabelMe\replacewords');

dirList = dir(HA);
for n = 3:length(dirList)
   dirPath = strcat(HA,'\',dirList(n).name);
   fileList = filenames(dirPath,'xml',2);
   for i = 1:length(fileList)
       [annotation, img] = LMread(fullfile(dirPath, fileList{i}), HI);
       [h w] = size(img);
       NI = h * w;
       if isfield(annotation,'object')
           No_objects = length(annotation.object);
           ind = 0;
           objName = '';
           for k = 1:No_objects
               if annotation.object(k).deleted == '0'
                   Obj =
cell2mat(strtrim(NI_replacewords(removestopwords(annotation.object(k).name),r
eplacewords)));
                   if ~isempty(Obj)
                       ind = ind + 1;
                       objName{ind} = Obj;
                   end
               end
           end

           % Creating an unique Object name list sorting with ascending order
           UobjName = unique(sort(objName));

           % Calculate Semantic Intensity of each object
           ind = 0;
           for j = 1:length(UobjName)
               SSI = 0;
               count = 0;
               for k = 1:No_objects
                   if
strcmpi(strtrim(NI_replacewords(removestopwords(annotation.object(k).name),re
placewords)),UobjName(j))
```

```
                        [X,Y] = getLMpolygon(annotation.object(k).polygon);
                        SI = polyarea(X,Y)/NI;
                        SSI = SSI + SI;
                        count = count + 1;
                    end
                end
                if SSI > 0
                    ind = ind + 1;
                    new_annotation.filename = annotation.filename;
                    new_annotation.folder = annotation.folder;
                    new_annotation.object(ind).id = ind-1;
                    new_annotation.object(ind).name = UobjName(j);
                    new_annotation.object(ind).SI = SSI;
                    new_annotation.object(ind).count = count;
                    disp(strcat(fileList{i},',  ',num2str(j),',
',UobjName(j)));
                end
            end
        end
            % Generating structure for the new files and then store them in
XML
            % format
            v.annotation = new_annotation;
            uwriteXML(fullfile(nHA,annotation.folder,fileList{i}),v);
            clear new_annotation;
   end
end
disp(' Unique Object name operation completed');
```

1.6. XML Re-Writter

```
function uwriteXML(filename, v)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% expand polygon for compatibility
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xml = struct2xml(v);

% Open file
fid = fopen(filename,'w');
fwrite(fid, xml, 'char');
% fprintf(fid, xml);
% Close file
fclose(fid);
```

1.7. Remove Redundancy from the Annotation

```
%% Setting of root folders
clear all;
HOMEIMAGES = 'F:\LabelMeDB\Images\'; % Source folder of the images
```

```matlab
HOMEANNOTATIONS = 'F:\LabelMeDB\Annotations\'; % Source folder of the
annotated xml files
newHomeAnnotation = 'F:\LabelMeDB\Annotations3\'; % Target folder to store
updated annotation files

%%

    dirPath = strcat(HOMEANNOTATIONS,'05june05_static_indoor');
    fn = 'p1010847.xml';

    % constructing the file path
    NI_filename = fullfile(dirPath,fn);

    % reading the annotation and image from the target folders and then
    % making changes according to the requirements as per descirption
    % below
    [annotation, img] = LMread(NI_filename, HOMEIMAGES);
    figure
    LMplot(annotation, img)
    No_objects = length(annotation.object);

    nr = annotation.imagesize.nrows;
    nc = annotation.imagesize.ncols;
    mi = str2double(nc) * str2double(nr);
    new_annotation = annotation;
    new_annotation.object = '';
    objName = {200};
    for k = 1:No_objects
        objName{k} = annotation.object(k).name;
    end
    clear k;

    % Creating an unique Object name list sorting with ascending order
    UobjName = unique(objName);
    UNo_objects = length(UobjName);

%%

    for j = 1:UNo_objects
        SI = 0;
        SSI = 0;
        count = 0;
        try
           irfan = '';
            for k = 1:length(objName)
                if strcmpi(annotation.object(k).name,UobjName(j))
                    [X,Y] = getLMpolygon(annotation.object(k).polygon);
                    SI = polyarea(X,Y)/mi;
                    SSI = SSI + SI;
                    count = count + 1;
                end
            end
            clear k;
        catch M
          irfan = 'error';
```

```
        end
         ASI = SSI / count;
         new_annotation.object(j).ID = j;
         new_annotation.object(j).name = strcat(UobjName(j),'
(',num2str(ASI),')');
         new_annotation.object(j).area = ASI;
    end
   %%
    nfilename = strcat(newHomeAnnotation,new_annotation.folder,'\',fn);
    v.new_annotation = new_annotation;
    NI_writeXML(nfilename,v);



%%
% Objects instances in a given image
No_Objects = length(annotation.object);
count = 0;
%%
fn = 'p1010847.xml';
objectName = {};
for n = 1:length(annotation.object)
    objectName{n} = annotation.object(n).name;
end
%%
UObjName = unique(objectName);

for i = 1:length(UObjName)
ASI = 0;
SSI = 0;
SI = 0;
count = 0;
for n = 1:length(objectName)
    if strcmpi(annotation.object(n).name,'grille') %UObjName(i))
        [X,Y] = getLMpolygon(annotation.object(n).polygon);
        SI = polyarea(X,Y)/10000;
        SSI = SSI + SI;
        count = count + 1;
    end
end
ASI = SSI / count;
end
```

a. Supporting Function: Filenames

```
function fileList = filenames(HOME,type, flage);
%
% Return list of the folders using MS-DOS commands
%
% folder = folderlist(HOME, type, flage)
%   folder = cell array
%   type = filetype .i.e. xml, docx, html
%   flage = 1 (file extraction of files from all subfolders)
%           2 (file extraction from the specified folder)
```

```
%
%


% Checking for input arguments
if nargin ~= 3
    fileList = 'Irfan';
    error('NI_filenames, Wrong number of input arguments')
end

try
if (flage == 1)

    % Extractions of folders from the varargin
    dirList = dir(HOME);

    % Extraction of files from the varargin
    j = 1;
    for n = 3:length(dirList)
        filetype = strcat(HOME, dirList(n).name,'\*.',type);
        dirContent = dir(filetype);

        % Extraction files from the structure
        for k = 1:length(dirContent)
            files{k} = dirContent(k).name;
        end
    fileList{j} = files;
    j = j + 1;
    end

elseif (flage == 2)
    filetype = strcat(HOME,'\*.',type);
    dirContent = dir(filetype);

    % Extraction files from the structure
    for k = 1:length(dirContent)
        fileList{k} = dirContent(k).name;
    end
   % fileList{1} = files;

else
    fileList = 'Irfan';
    error('NI-LMfilenames --> flage should be 1 or 2');

end;
catch m
    fileList = 'Irfan';
    error('NI-LMfilenames --> no files exists');
end;
```

b. Word_Replacement function

```
function name = NI_replacewords(name, repwords)
% replace words by using
if nargin < 2
    % load list of replacewords
    fid = fopen('replacewords.txt');
    C = textscan(fid,'%s');
    fclose(fid);
    repwords = C{1};
end
name = strrep(name,repwords,'');
end
```

c. Semantic Similarity Calculation

```
%% Semantic Similarity Code of C# (.dll file) in Matlab

% Loading the DLL library from the specified path
try
    NET.addAssembly('F:\SAR.dll');

    % Creating instance of the Class
    P = SAR.WordNet.SimSem;
catch M
    clc;
    error('Problem in Library Loading...');
end
clear M;
% Using methods from the class
P.SimSim('Thanks','Good');
t = 'car';
s = 'vehicle';
P.SimSim(cell2mat(t), cell2mat(s));

clear P t s M;

%% Using with other terms
```

1.8. Annotation Refinement

```
function DB = NI_Struct_Annotation(D, HI, HA)
% This function give us a unqiue name of the object in the Database file,
% the out argument is the unique object name structure for the further
% processing
%

   % extracts the object name for processing of stopwords and
   % replacewords

  % Load stopwords and replacewords list
   load('D:\Research\LabelMe\stopwords');
   load('D:\Research\LabelMe\replacewords');
```

```
for n = 1:length(D)
    NIR = 1;
    try
        fn =
fullfile(HA,D(n).annotation.folder,strrep(D(n).annotation.filename,'.jpg','.x
ml'));
        [annotation, img] = LMread(fn, HI);
        [h w d] = size(img);
        NI = h * w;
    catch m2
        NIR = 0;
        error('LMread error');
    end
    A = annotation;
    annotation.object='';
    clear m2;
    objName = '';

   if ismember(A,'object')
       for p = 1:length(A.object)
           objName{p} =
NI_porterStemmer(cell2mat(NI_replacewords(removestopwords(A.object(p).name,st
opwords),replacewords)));
       end
       clear p;
   else
       NIR = 0;
       error('Image not annotated yet...');
   end

   % sorting and unique operations
   sobjName = sort(objName);

   % removing the blank objects name
   h = 1;
   snobjName = '';

   if NIR == 1
        for p = 1: length(sobjName)
            if length(sobjName{p})>0
                snobjName{h} = (sobjName{p});
                h = h + 1;
            end
        end
        clear p h;
   end

   uobjName = unique(snobjName);

   % Totaling the object area
   if NIR == 1
       for k = 1:length(uobjName)
           TSI = 0;
```

```
            % Calculate Semantic Intensity of each object
            for j = 1:length(A.object)
                flage = 0;
                if
~(strcmpi((uobjName{k}),(NI_replacewords(removestopwords(A.object(j).name,sto
pwords),replacewords))))
                    [X,Y] = getLMpolygon(A.object(j).polygon);
                    Area = polyarea(X,Y)/NI;
                    flage = 1;
                end

                if flage == 1
                    TSI = TSI + Area;
                end
            end

            annotation.object(k).id = k-1;
            annotation.object(k).name = strcat(uobjName{k},'
(',num2str(TSI),')');
            annotation.object(k).SI = TSI;
        end
    end

    % Generating structure for the new files and then store them in XML
    % format
    DB(n).annotation = annotation;
    clc;
    disp(strcat(num2str(n),' -- records are processed...'));
end
end

%%
```

1.9. Uniquness in the Annotation

```
function NI_uAnnotation(HOMEIMAGES, HOMEANNOTATIONS,uHomeAnnotation)
% This function give us a unqiue name of the object in the annotation file
% storing in another location. In this case the new annotation folder is
% the Annotations3. The arguments are as
%
% HOMEIMAGES : Its the root path for the images
% HOMEANNOTATIONS : Its the root path for the original annotations

clc;

% Load stopwords and replacewords list
load('D:\Research\LabelMe\stopwords');
load('D:\Research\LabelMe\replacewords');

dirList = dir(HOMEANNOTATIONS);

for n = 3:length(dirList)
    dirPath = strcat(HOMEANNOTATIONS,dirList(n).name);
```

```
    try
        NIR = 1;
        fileList = NI_filenames(dirPath,'xml',2);

        for i = 1:length(fileList)
            field = {'file'};
            fileName = cell2struct(fileList(i),field);
            fn = fileName.file;
            NI_filename = fullfile(dirPath,fn);

            %NI_filename =
strcat(HOMEANNOTATIONS,dirList(dl).name,'\',fileName.file(i));
            try
                [annotation, img] = LMread(NI_filename, HOMEIMAGES);
                [h w d] = size(img);
                NI = h * w;
            catch m2
                error('LMread error');
                NIR = 0;
            end




            A = annotation;
            annotation.object='';

            % extracts the object name for processing of stopwords and
            % replacewords


            objName = '';

            try
                for p = 1:length(A.object)
                    objName{p} =
cell2mat(NI_replacewords(removestopwords(A.object(p).name,stopwords),replacew
ords));
                end
                clear p;
            catch m1
               error('Image not annotated yet...');
               NIR = 0;
            end

            % sorting and unique operations
            sobjName = sort(objName);

            % removing the blank objects name
            h = 1;
            snobjName = '';

           if NIR == 1
               for p = 1: length(sobjName)
                   if length(sobjName{p})>0
```

```
                    snobjName{h} = (sobjName{p});
                    h = h + 1;
                end
            end
            clear p h;
        end

         uobjName = unique(snobjName);

         % Totaling the object area
         if NIR == 1
             for k = 1:length(uobjName)
                 TSI = 0;
                 % Calculate Semantic Intensity of each object
                 for j = 1:length(A.object)
                     flage = 0;

                     if
~(strcmpi((uobjName{k}),(NI_replacewords(removestopwords(A.object(j).name,sto
pwords),replacewords))))
                         [X,Y] = getLMpolygon(A.object(j).polygon);
                         Area = polyarea(X,Y)/NI;
                         flage = 1;
                     end

                     if flage == 1
                         TSI = TSI + Area;
                     end
                 end

                 annotation.object(k).id = k-1;
                 annotation.object(k).name = strcat(uobjName{k},'
(',num2str(TSI),')');
                 annotation.object(k).SI = TSI;
             end
         end
         % Generating structure for the new files and then store them in
XML
         % format
         nfilename = strcat(uHomeAnnotation,annotation.folder,'\',fn);
         v.annotation = annotation;
         NI_writeXML(nfilename,v);

         clc;
         disp(strcat(num2str(i),'/',num2str(n-2),' -- files / folder
processed...'));
     end
   catch m
   end
end
end
```

1.10. List of the stopwords

```
group
of
aszxaszx
ccccccccccc
sideview
walking
lowres
dark
sitting
gray
red
blue
white
brown
black
side
frontal
part
behind
crop
rear
back
front
left
right
occluded
spinning
the
in
a
view
big
whole
partial
az0deg
az30deg
az60deg
az90deg
az120deg
az150deg
az180deg
az210deg
az240deg
az270deg
az300deg
az330deg
az360deg
1
2
3
4
111
10-207
```

```
55
adding
aibo
unit
*x
```

1.11. DCS Annotation structure to LabelMe XML format function

```
%% Using DCS Dataset
% Path = D:\Research\Example\DCS -- Images
% File is loaded manually and then Performing the following operations
clear all;
load('D:\Research\Datasets\DCS Datasets\DCS -- Annotation\DCS.mat');
for i = 1:length(BA)
    [a b] = strtok(BA(i));
    [c d] = strtok(b);
    [e f] = strtok(d);
    [g h] = strtok(f);
    [k l] = strtok(h);
    [m n] = strtok(l);
    BA1(i,1) = a;
    BA1(i,2) = c;
    BA1(i,3) = e;
    BA1(i,4) = g;
    BA1(i,5) = k;
    BA1(i,6) = m;
end
clear i a b c d e f g h k l m n BA;

% now compiling XML files
folderPath=inputdlg('Enter path of the folder: ');
for i = 1:length(BA1)
    DCS(i).annotation.filepath =
cell2mat(strcat(cell2mat(folderPath),BA1(i,1)));
    for j = 2:6
         if length(cell2mat(BA1(i,j)))>0
            DCS(i).annotation.object(j-1).name = cell2mat(BA1(i,j));
        end
    end
end
clear i j;
%%
% structure 2 xml annotation form
fileName = fullfile(inputdlg('Enter file name: '));
v.DCS = DCS;
NI_writeXML(cell2mat(fileName), v);
clear v fileName;
```

1.12. Unique Concept Semantic Intensity Calculation

```
%% Setting of root folders
clc;
clear all;
```

```
HomeImage = 'D:\LabelMeDB\Images\'; % Source folder of the images
HomeAnnotation = 'D:\LabelMeDB\Annotations\'; % Source folder of the
annotated xml files
newHomeAnnotation = 'D:\LabelMeDB\Annotations2\'; % Target folder to store
objects with their SI
uHomeAnnotation = 'D:\LabelMeDB\Annotations3\';  % Target folder to store
unqiue object with their total SI

%% Performing unique object SI calculation
clc;
disp(' Calculationg unqiue object and SI for the objects...');
NI_uAnnotation(HOMEIMAGES, HOMEANNOTATIONS, uHomeAnnotation);

%% Database Creation from the corpus
clc;
disp(' ');
disp(' Database --> XML in progress...');
D = NI_Database(uHomeAnnotation);
disp('=============================================');
disp(' Database XML --> Structure completed...');


%% Extracting all object names from the corpus
clc;
k = 1;
objName = '';
load('D:\Research\LabelMe\stopwords');
load('D:\Research\LabelMe\replacewords');
for i = 1: length(D)
    if isfield(D(i).annotation,'object')
        for j = 1:length(D(i).annotation.object)
            try
                objName{k}=
cell2mat(NI_replacewords(removestopwords(D(i).annotation.object(j).name,stopw
ords),replacewords));
                k = k+1;
            catch m
                error(strcat('Errors occure at :',num2str(k)));
            end
        end
        disp(strcat(num2str(i),' - images are processed'));
    end
end
clear i j k m;

%% Re-arranging the objName
% check the objName whether its in cell form or not and then use it
% accordingly

for i = 1: length(objName)
    cobjName{i} = cell2mat(objName{i});
end
%% sorting all the objects name and then extracts unqiue from them

sobjName = sort(uobjName);
```

```
%%
uobjName = unique(objName);
%% Quering Database for specific Object
clc;
t = input(' Enter name of the object to be queried: ','s');
Dq = LMquery(D,'object.name',t);
clear t;

%% Displaying Query results
for n = 1: length(Dq)
    fn =
fullfile(uHomeAnnotation,Dq(n).annotation.folder,strrep(Dq(n).annotation.file
name,'.jpg','.xml'));
    [annotation img] = LMread(fn, HOMEIMAGES);
    objName = '';
    for i = 1: length(annotation.object)
        objName = strcat(objName,', ', annotation.object(i).name);
    end
    figure;
    imshow(img);
    title(objName);
end
clear n fn i;% objName;
```

2. C# Programming Codes

2.1. Source Code: Main Program

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace Irfan
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Main());
        }
    }
}
```

2.2. Source Code: Main Interface

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MLApp;

namespace Irfan
{
    public partial class Main : Form
    {
        public Main()
        {
            InitializeComponent();
        }

        #region Main Declaration

        public static string[] Concept = new string[50];
        public static string Conceptword;
        public  struct LConceptSS
        {
            private string sConcept;
            public string Concept
            {
                get
                {
                    return sConcept;
                }
                set
                {
                    sConcept = value;
                }
            }

            private double sSS;
            public double SS
            {
                get
                {
                    return sSS;
                }
                set
                {
                    sSS = value;
                }
            }
        }
        public static LConceptSS[] ConceptSS = new LConceptSS[Concept.Length];
        public static DataTable GridData(double f )
        {
```

```csharp
    // -------------------- Main GridView ----------------------//
    DataTable Pir = new DataTable("ConceptList");
    DataColumn Concept = new DataColumn("Concept");
    DataColumn SS = new DataColumn("SS");
    Pir.Columns.Add(Concept);
    Pir.Columns.Add(SS);
    DataRow newRow;
    // -------------------- Main GridView ----------------------//
    for (int i = 0; i <= Main.ConceptSS.Length - 1; i++)
    {
        if (Main.ConceptSS[i].Concept != null && Main.ConceptSS[i].SS >= f)
        {
            newRow = Pir.NewRow();
            newRow["Concept"] = Main.ConceptSS[i].Concept;
            newRow["SS"] = Main.ConceptSS[i].SS;
            Pir.Rows.Add(newRow);
        }
    }
    return Pir;
}
#endregion

//public static string[] ConceptSS = new string[Concept.Length];

private void button1_Click(object sender, EventArgs e)
{
    ConceptNet.FileOptionsForm fof = new ConceptNet.FileOptionsForm();
    fof.Show();
}

private void button2_Click(object sender, EventArgs e)
{
    ConceptNet.ConceptExtraction CE = new ConceptNet.ConceptExtraction();
    CE.Show();
}

private void button1_Click_1(object sender, EventArgs e)
{
    WordNet.SimSem S = new Irfan.WordNet.SimSem();
    S.Show();
}

private void button3_Click(object sender, EventArgs e)
{
    Irfan.WordNet.LexiconSample LS = new Irfan.WordNet.LexiconSample();
    LS.Show();
}

private void button4_Click(object sender, EventArgs e)
{
    Matlab.Matlab M = new Irfan.Matlab.Matlab();
    M.Show();
}

private void button5_Click(object sender, EventArgs e)
{
    MainDataGridView.DataSource = GridData(0.00);
}
```

```
    }
}
```

2.3. Supporting tools for the research:

We have used the following supporting code for WordNet, ConceptNet and Montylingua for the research purpose, all these code are available openly for the research purposes. Next we will describe the supporting tools one/one

a. WordNet Supporting tools:

For WordNet support, we have selected the tools from the code project written by Tunaah, for sentence similarity, word ambiguity and semantic similarity among the words. The functions that are used during the research process are

i. ISimilarity.cs
ii. Relatedness.cs
iii. SentenceSimilarity.cs
iv. SimilarGenerator.cs
v. WordSenseDisambiguity.cs
vi. WordSimilarity.cs
vii. Matcher.BipartiteMatcher.cs
viii. Matcher.HeuristicMatcher.cs
ix. TextHelper.Acronym.cs
x. TexHelpre.ExtOverlapCounter.cs
xi. TextHelper.StopWordsHandler.cs
xii. TextHelper.Tokeniser.cs

These function are jointly used to calculate the semantic similarity among the words. The source code for the semantic similarity are

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using WordsMatching;
```

```
namespace Irfan.WordNet
{
    public partial class SimSem : Form
    {
        public SimSem()
        {
            InitializeComponent();
            Wnlib.WNCommon.path = "C:\\Program Files\\WordNet\\2.1\\dict\\";
            tbOrigConcept.Text = Main.Conceptword;
        }

        private void btnScore_Click(object sender, EventArgs e)
        {
            SentenceSimilarity semsim = new SentenceSimilarity();
            txt3.Text = "";
            txt3.Text += semsim.GetScore(txt1.Text, txt2.Text);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SentenceSimilarity semsim = new SentenceSimilarity();

            // Calculating Semantic Similarity and store the result in local structure
            for (int i = 0; i <= Main.ConceptSS.Length - 1; i++)
            {
                if (Main.ConceptSS[i].Concept != null && Main.ConceptSS[i].Concept!=
Main.Conceptword)
                {
                    Main.ConceptSS[i].SS =
semsim.GetScore(Main.Conceptword,Main.ConceptSS[i].Concept);
                }
            }
            dgConcept.DataSource = Main.GridData(0.00);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            dgFilterConcept.DataSource = Main.GridData(Convert.ToDouble(tbth.Text));
        }
    }
}
```

b. ConceptNet:

The Code for this module is taken from the code project openly available for research purposes; we have modified the coder as per our requirements. The snapshot of the source code is under. These code are written for ConceptNet 2.1 version.

Function: Handling the ConceptExtraction

```
////////////////////////////////
///Form1.cs - version 0.01412006.0rc4
///BY DOWNLOADING AND USING, YOU AGREE TO THE FOLLOWING TERMS:
///Copyright (c) 2006 by Joseph P. Socoloski III
///LICENSE
///If it is your intent to use this software for non-commercial purposes,
///such as in academic research, this software is free and is covered under
///the GNU GPL License, given here: <http://www.gnu.org/licenses/gpl.txt>
///
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using ConceptNetUtils;
using MLApp;
using StringProcessing;

namespace Irfan.ConceptNet
{
	/// <summary>
	/// Summary description for Form1.
	/// </summary>
	public class ConceptExtraction : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Label label1;
		private System.Windows.Forms.TextBox tbWord;
		private System.Windows.Forms.Label label2;
		private System.Windows.Forms.ComboBox cbRelationshipTypes;
		private System.Windows.Forms.Label label3;
		private System.Windows.Forms.TextBox tbMAXResults;
		private System.Windows.Forms.CheckBox blCreateOutputFile;
		private System.Windows.Forms.GroupBox groupBox1;
		private System.Windows.Forms.TextBox txtOut;
         private System.Windows.Forms.Button btSearch;
		private System.ComponentModel.IContainer components;

		//
         //Editing by Irfan
         //
		string TextOutputFilename = "defaultname";
         string s;



         //Initialize ConceptNetUtils
		ConceptNetUtils.Search CNSearch = new ConceptNetUtils.Search();
		ConceptNetUtils.FoundList CNFoundList = new ConceptNetUtils.FoundList();
		ConceptNetUtils.Misc CNMisc = new ConceptNetUtils.Misc();
		private System.Windows.Forms.Button btSortbyf;
         private System.Windows.Forms.Button btSortbyi;
         private BindingSource mLAppClassBindingSource;
         private Panel panel1;
```

```
        private TextBox tbConceptArray;
        private Panel panel2;
        private Button button2;
        private PictureBox pictureBox1;
              ArrayList ALFoundList = new ArrayList();


              public  ConceptExtraction()
              {
                     //
                     // Required for Windows Form Designer support
                     //
                     InitializeComponent();

                     //
                     // TODO: Add any constructor code after InitializeComponent call
                     //
              }

              /// <summary>
              /// Clean up any resources being used.
              /// </summary>
              protected override void Dispose( bool disposing )
              {
                     if( disposing )
                     {
                            if (components != null)
                            {
                                   components.Dispose();
                            }
                     }
                     base.Dispose( disposing );
              }

              #region Windows Form Designer generated code
              /// <summary>
              /// Required method for Designer support - do not modify
              /// the contents of this method with the code editor.
              /// </summary>
              private void InitializeComponent()
              {
            this.components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(ConceptExtraction));
            this.label1 = new System.Windows.Forms.Label();
            this.tbWord = new System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.cbRelationshipTypes = new System.Windows.Forms.ComboBox();
            this.label3 = new System.Windows.Forms.Label();
            this.tbMAXResults = new System.Windows.Forms.TextBox();
            this.blCreateOutputFile = new System.Windows.Forms.CheckBox();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.btSortbyi = new System.Windows.Forms.Button();
            this.txtOut = new System.Windows.Forms.TextBox();
            this.btSortbyf = new System.Windows.Forms.Button();
            this.btSearch = new System.Windows.Forms.Button();
            this.mLAppClassBindingSource = new
System.Windows.Forms.BindingSource(this.components);
```

```
            this.panel1 = new System.Windows.Forms.Panel();
            this.button2 = new System.Windows.Forms.Button();
            this.tbConceptArray = new System.Windows.Forms.TextBox();
            this.panel2 = new System.Windows.Forms.Panel();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.groupBox1.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.mLAppClassBindingSource)).BeginInit();
            this.panel1.SuspendLayout();
            this.panel2.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
            this.SuspendLayout();
            //
            // label1
            //
            this.label1.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label1.Location = new System.Drawing.Point(8, 9);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(344, 27);
            this.label1.TabIndex = 2;
            this.label1.Text = "Type Your Subject Here (one word only):";
            //
            // tbWord
            //
            this.tbWord.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.tbWord.Location = new System.Drawing.Point(458, 9);
            this.tbWord.Name = "tbWord";
            this.tbWord.Size = new System.Drawing.Size(346, 27);
            this.tbWord.TabIndex = 3;
            this.tbWord.TextChanged += new System.EventHandler(this.tbWord_TextChanged);
            this.tbWord.Leave += new System.EventHandler(this.tbWord_Leave);
            //
            // label2
            //
            this.label2.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label2.Location = new System.Drawing.Point(8, 55);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(432, 27);
            this.label2.TabIndex = 4;
            this.label2.Text = "What relationship type do you which to search for?";
            //
            // cbRelationshipTypes
            //
            this.cbRelationshipTypes.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.cbRelationshipTypes.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.cbRelationshipTypes.ImeMode = System.Windows.Forms.ImeMode.NoControl;
            this.cbRelationshipTypes.Items.AddRange(new object[] {
            "K-Lines: ConceptuallyRelatedTo",
            "K-Lines: ThematicKLine",
            "K-Lines: SuperThematicKLine",
            "All K-Lines",
            "Things: IsA",
            "Things: PartOf",
```

```
            "Things: PropertyOf",
            "Things: DefinedAs",
            "Things: MadeOf",
            "All Things",
            "Spatial: LocationOf",
            "Events: SubeventOf",
            "Events: PrerequisiteEventOf",
            "Events: First-SubeventOf",
            "Events: LastSubeventOf",
            "All Events",
            "Causal: EffectOf",
            "Causal: DesirousEffectOf",
            "All Causal",
            "Affective: MotivationOf",
            "Affective: DesireOf",
            "All Affective",
            "Functional: CapableOfReceivingAction",
            "Functional: UsedFor",
            "All Functional",
            "Agents: CapableOf",
            "All (Returns all results with word)"});
            this.cbRelationshipTypes.Location = new System.Drawing.Point(458, 55);
            this.cbRelationshipTypes.Name = "cbRelationshipTypes";
            this.cbRelationshipTypes.RightToLeft = System.Windows.Forms.RightToLeft.No;
            this.cbRelationshipTypes.Size = new System.Drawing.Size(346, 26);
            this.cbRelationshipTypes.TabIndex = 5;
            this.cbRelationshipTypes.SelectedIndexChanged += new
System.EventHandler(this.cbRelationshipTypes_SelectedIndexChanged);
            //
            // label3
            //
            this.label3.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label3.Location = new System.Drawing.Point(8, 111);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(460, 26);
            this.label3.TabIndex = 6;
            this.label3.Text = "Set the Maximum number of results to display (1-999):";
            //
            // tbMAXResults
            //
            this.tbMAXResults.Font = new System.Drawing.Font("Arial", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.tbMAXResults.Location = new System.Drawing.Point(458, 111);
            this.tbMAXResults.Name = "tbMAXResults";
            this.tbMAXResults.Size = new System.Drawing.Size(106, 30);
            this.tbMAXResults.TabIndex = 7;
            this.tbMAXResults.Text = "50";
            this.tbMAXResults.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center;
            //
            // blCreateOutputFile
            //
            this.blCreateOutputFile.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.blCreateOutputFile.Location = new System.Drawing.Point(16, 141);
            this.blCreateOutputFile.Name = "blCreateOutputFile";
            this.blCreateOutputFile.Size = new System.Drawing.Size(760, 27);
```

```
            this.blCreateOutputFile.TabIndex = 8;
            this.blCreateOutputFile.Text = "Create a text file with results";
            //
            // groupBox1
            //
            this.groupBox1.BackColor = System.Drawing.Color.LightSteelBlue;
            this.groupBox1.Controls.Add(this.btSortbyi);
            this.groupBox1.Controls.Add(this.txtOut);
            this.groupBox1.Controls.Add(this.btSortbyf);
            this.groupBox1.Location = new System.Drawing.Point(10, 178);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(794, 356);
            this.groupBox1.TabIndex = 9;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Results...";
            //
            // btSortbyi
            //
            this.btSortbyi.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btSortbyi.Location = new System.Drawing.Point(422, 314);
            this.btSortbyi.Name = "btSortbyi";
            this.btSortbyi.Size = new System.Drawing.Size(288, 26);
            this.btSortbyi.TabIndex = 12;
            this.btSortbyi.Text = "Sort by i (# of times inferred)";
            this.btSortbyi.Click += new System.EventHandler(this.btSortbyi_Click);
            //
            // txtOut
            //
            this.txtOut.BackColor = System.Drawing.Color.White;
            this.txtOut.Font = new System.Drawing.Font("Lucida Console", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.txtOut.Location = new System.Drawing.Point(20, 22);
            this.txtOut.MaxLength = 992767;
            this.txtOut.Multiline = true;
            this.txtOut.Name = "txtOut";
            this.txtOut.ReadOnly = true;
            this.txtOut.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
            this.txtOut.Size = new System.Drawing.Size(744, 285);
            this.txtOut.TabIndex = 10;
            //
            // btSortbyf
            //
            this.btSortbyf.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btSortbyf.Location = new System.Drawing.Point(96, 314);
            this.btSortbyf.Name = "btSortbyf";
            this.btSortbyf.Size = new System.Drawing.Size(260, 26);
            this.btSortbyf.TabIndex = 11;
            this.btSortbyf.Text = "Sort by f (# of utterances)";
            this.btSortbyf.Click += new System.EventHandler(this.btSortbyf_Click);
            //
            // btSearch
            //
            this.btSearch.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.btSearch.Location = new System.Drawing.Point(582, 97);
            this.btSearch.Name = "btSearch";
```

```
            this.btSearch.Size = new System.Drawing.Size(222, 46);
            this.btSearch.TabIndex = 10;
            this.btSearch.Text = "Search";
            this.btSearch.Click += new System.EventHandler(this.btSearch_Click);
            //
            // mLAppClassBindingSource
            //
            this.mLAppClassBindingSource.DataSource = typeof(MLApp.MLAppClass);
            //
            // panel1
            //
            this.panel1.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(192)))),
((int)(((byte)(192)))), ((int)(((byte)(255)))));
            this.panel1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
            this.panel1.Controls.Add(this.button2);
            this.panel1.Controls.Add(this.tbConceptArray);
            this.panel1.Location = new System.Drawing.Point(930, 56);
            this.panel1.Name = "panel1";
            this.panel1.Size = new System.Drawing.Size(316, 551);
            this.panel1.TabIndex = 13;
            //
            // button2
            //
            this.button2.Location = new System.Drawing.Point(18, 21);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(278, 83);
            this.button2.TabIndex = 14;
            this.button2.Text = "Concept(s) Purification";
            this.button2.UseVisualStyleBackColor = true;
            this.button2.Click += new System.EventHandler(this.button2_Click);
            //
            // tbConceptArray
            //
            this.tbConceptArray.BackColor =
System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(224)))),
((int)(((byte)(192)))));
            this.tbConceptArray.Location = new System.Drawing.Point(18, 111);
            this.tbConceptArray.Multiline = true;
            this.tbConceptArray.Name = "tbConceptArray";
            this.tbConceptArray.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
            this.tbConceptArray.Size = new System.Drawing.Size(278, 423);
            this.tbConceptArray.TabIndex = 13;
            //
            // panel2
            //
            this.panel2.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(192)))),
((int)(((byte)(192)))), ((int)(((byte)(255)))));
            this.panel2.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
            this.panel2.Controls.Add(this.btSearch);
            this.panel2.Controls.Add(this.groupBox1);
            this.panel2.Controls.Add(this.blCreateOutputFile);
            this.panel2.Controls.Add(this.tbMAXResults);
            this.panel2.Controls.Add(this.tbWord);
            this.panel2.Controls.Add(this.label3);
            this.panel2.Controls.Add(this.cbRelationshipTypes);
            this.panel2.Controls.Add(this.label2);
            this.panel2.Controls.Add(this.label1);
            this.panel2.Location = new System.Drawing.Point(82, 56);
```

```
            this.panel2.Name = "panel2";
            this.panel2.Size = new System.Drawing.Size(818, 551);
            this.panel2.TabIndex = 14;
            //
            // pictureBox1
            //
            this.pictureBox1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.pictureBox1.Image =
((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));
            this.pictureBox1.Location = new System.Drawing.Point(0, 0);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(1346, 668);
            this.pictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
            this.pictureBox1.TabIndex = 15;
            this.pictureBox1.TabStop = false;
            //
            // ConceptExtraction
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(6, 15);
            this.ClientSize = new System.Drawing.Size(1346, 668);
            this.Controls.Add(this.panel2);
            this.Controls.Add(this.panel1);
            this.Controls.Add(this.pictureBox1);
            this.Name = "ConceptExtraction";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "Concept Extraction";
          //  this.Load += new System.EventHandler(this.Form1_Load);
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.mLAppClassBindingSource)).EndInit();
            this.panel1.ResumeLayout(false);
            this.panel1.PerformLayout();
            this.panel2.ResumeLayout(false);
            this.panel2.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
            this.ResumeLayout(false);

             }
             #endregion

             /// <summary>
             /// The main entry point for the application.
             /// </summary>


             /////////////////////////////////////ConceptNet Demo App Methods
Below////////////////////////////////////////////
             private void btSearch_Click(object sender, System.EventArgs e)
             {
                    Cursor.Current = Cursors.WaitCursor;

                    //Reset txtOut
                    txtOut.Text = "";
                    string searchresultstodisplay ="";

                    string demofolderpath = Application.StartupPath;
```

```
                    //Set/Initialize Predicatefile variables for the class library after
loading them from an XML file.
                    CNSearch.XMLLoadFilePaths("D:\\Visual Studio
2010\\Irfan\\References\\Settings.xml");

                    //if there is a word in the Textbox then it's ok to start search...
                    if(tbWord.Text != "")
                    {
                           try
                           {
                                  //Make sure tbWord.Text is lowercase
                                  tbWord.Text = tbWord.Text.ToLower();

                                  //Reset List(s) to null.
                                  CNSearch.Clear();
                                  CNFoundList.Reset();
                                  ALFoundList.Clear();

                                  //If checked in one of the , Search them...
                                  //Preform Search using ConceptNetUtil Class Library
                                  CNSearch.XMLSearchForChecked("D:\\Visual Studio
2010\\Irfan\\References\\Settings.xml", tbWord.Text.Trim(),
CNMisc.RemoveCategoryString(cbRelationshipTypes.Text),
Convert.ToInt32(tbMAXResults.Text), blCreateOutputFile.Checked, demofolderpath + @"\" +
TextOutputFilename);

                                  //***Copy the
ConceptNetUtils.SearchResultsList.FoundList so not to lose scope***
                                  int numberoflines = CNSearch.GetTotalLineCount();
                                  for(int i = 0; i < numberoflines ; i++)
                                  {
                                         //Copy into a global ArrayList
                                         ALFoundList.Add(CNSearch.GetFoundListLine(i));
                                         //Copy into a global CNFoundList
                                         CNFoundList[i] = CNSearch.GetFoundListLine(i);
                                  }

                                  System.Collections.IEnumerator myEnumerator =
ALFoundList.GetEnumerator();
                                  while ( myEnumerator.MoveNext() )
                                         searchresultstodisplay +=
myEnumerator.Current.ToString() + "\r\n";

                                  //Now display in txtOut
                                  int totalfound = CNSearch.GetTotalLineCount();
                    // Edit by Irfan
                  //  searchresultstodisplay += ("---------------------- Done -----------
-----------\r\n");
                                  //searchresultstodisplay +=
(Convert.ToString(totalfound) + " " + cbRelationshipTypes.Text +" Found.");

                    txtOut.Text = searchresultstodisplay;
                                  txtOut.Update();
                           }
                           catch  (Exception ex)
                           {
                                  //tbWord.Text did not have a subject and/or
```

```
                                //fileandpath may have been incorrect.
                                MessageBox.Show("Make sure you have a word typed in the
inputbox and \r\nMake sure you are pointing to the correct path for ConceptNet.\r\n" +
ex.Message);
                                searchresultstodisplay += "---------------------- Error
----------------------\r\n";
                                txtOut.Text = searchresultstodisplay;
                                txtOut.Update();
                        }
                }
                else
                {
                        MessageBox.Show("You must type in a word to perform a
search.");
                txtOut.Update();
                }

                //Create Text file if checked
                if(blCreateOutputFile.Checked)
                {
                        TextOutputFilename = tbWord.Text + "_" +
CNMisc.RemoveCategoryString(cbRelationshipTypes.Text) + ".txt";
                CNSearch.CreateTextFile("D:\\Visual Studio
2010\\Irfan\\References\\Concept Text\\" + TextOutputFilename);
                s = "D:\\Visual Studio 2010\\Irfan\\References\\Concept Text\\" +
TextOutputFilename;
                }

                Cursor.Current = Cursors.Default;
        }

        private void tbWord_TextChanged(object sender, System.EventArgs e)
        {
                //Create output file name string
                TextOutputFilename = tbWord.Text + "_" +
CNMisc.RemoveCategoryString(cbRelationshipTypes.Text) + ".txt";

                //Update to checkbox text
                blCreateOutputFile.Text = "Create a text file with results named: "
+ tbWord.Text + "_" + CNMisc.RemoveCategoryString(cbRelationshipTypes.Text) + ".txt";
                blCreateOutputFile.Update();
        }

        private void cbRelationshipTypes_SelectedIndexChanged(object sender,
System.EventArgs e)
        {
                cbRelationshipTypes.BeginUpdate();

                //If the form just loaded, do not change checkbox Text
                if(TextOutputFilename == "defaultname")
                {

                }
                else
                {
                        //Create output file name string
                        TextOutputFilename = tbWord.Text + "_" +
CNMisc.RemoveCategoryString(cbRelationshipTypes.Text) + ".txt";
```

```
                            //Update to checkbox text
                            blCreateOutputFile.Text = "Create a text file with results
named: " + tbWord.Text + "_" + CNMisc.RemoveCategoryString(cbRelationshipTypes.Text) +
".txt";
                            blCreateOutputFile.Update();
                     }

                     cbRelationshipTypes.Update();
                     cbRelationshipTypes.EndUpdate();
              }

              private void tbWord_Leave(object sender, System.EventArgs e)
              {
                     //Make sure tbWord.Text is lowercase
                     tbWord.Text = tbWord.Text.ToLower();
                     tbWord.Update();
              }

              private void btSortbyf_Click(object sender, System.EventArgs e)
              {
                     Cursor.Current = Cursors.WaitCursor;

                     //Create ArrayList to hold return sort results
                     ArrayList Listranked = new ArrayList();

                     //"Lift" the heaviest relationships to the top of the ArrayList
                     CNSearch.Sort_f(ALFoundList, out Listranked);

                     //Overwrite the old ALFoundList with the new ranking
                     ALFoundList = Listranked;

                     string searchresultstodisplay = "";

                     System.Collections.IEnumerator myEnumerator =
ALFoundList.GetEnumerator();
                            while ( myEnumerator.MoveNext() )
                                   searchresultstodisplay +=
myEnumerator.Current.ToString() + "\r\n";

                     //Now display in txtOut
                     searchresultstodisplay += ("---------------------- Done ------------
----------\r\n");
                     searchresultstodisplay += (Convert.ToString(ALFoundList.Count) + " "
+ cbRelationshipTypes.Text +" Found.");

                     txtOut.Text = searchresultstodisplay;
                     txtOut.Update();

                     Cursor.Current = Cursors.Default;
              }

              private void btSortbyi_Click(object sender, System.EventArgs e)
              {
                     Cursor.Current = Cursors.WaitCursor;

                     //Create ArrayList to hold return sort results
                     ArrayList Listranked = new ArrayList();
```

```
                    //"Lift" the heaviest relationships to the top of the ArrayList
                    CNSearch.Sort_i(ALFoundList, out Listranked);

                    //Overwrite the old ALFoundList with the new ranking
                    ALFoundList = Listranked;

                    string searchresultstodisplay = "";

                    System.Collections.IEnumerator myEnumerator =
ALFoundList.GetEnumerator();
                    while ( myEnumerator.MoveNext() )
                           searchresultstodisplay += myEnumerator.Current.ToString() +
"\r\n";

                    //Now display in txtOut
                    searchresultstodisplay += ("---------------------- Done ------------
----------\r\n");
                    searchresultstodisplay += (Convert.ToString(ALFoundList.Count) + " "
+ cbRelationshipTypes.Text +" Found.");

                    txtOut.Text = searchresultstodisplay;
                    txtOut.Update();

                    Cursor.Current = Cursors.Default;
              }

        private void button2_Click(object sender, EventArgs e)
        {
            // Irfan Editing this
            int a, h = 0;
            string st = txtOut.Text;
            Main.Conceptword = tbWord.Text;
            while (st.Length > 0)
            {
                try
                {
                    a = st.IndexOf('('); a++; st = st.Substring(a);
                    a = st.IndexOf('"'); a++; st = st.Substring(a);
                    a = st.IndexOf('"'); Main.ConceptSS[h++].Concept = st.Substring(0,
a); a++; st = st.Substring(a);
                    a = st.IndexOf('"'); a++; st = st.Substring(a);
                    a = st.IndexOf('"'); a++; st = st.Substring(a);
                    a = st.IndexOf('"'); a++; st = st.Substring(a);
                    a = st.IndexOf('"'); a++; st = st.Substring(a);
                    a = st.IndexOf(')'); a++; st = st.Substring(a);
                }
                catch
                {
                    break;
                }
            }
            tbConceptArray.Text = "";
            for (a = 0; a <= Main.ConceptSS.Length - 1; a++)
            {
                if (Main.ConceptSS[a].Concept != null && Main.Conceptword !=
Main.ConceptSS[a].Concept)
                {
```

```
                    tbConceptArray.Text += Main.ConceptSS[a].Concept + "\r\n";
                }
            }
        }
    }
}
```

c. Matlab:

As per requirement of the research, some of our work is perform in Matlab, while for some C# tool is used. We have used the utility MLApp for C# to call the Matlab function. Further, we have handle the Matlab function execution through threading process. The source code for different purpose perform in the Matlab are (for the Matlab function are giving under the head of Matlab code). The following are the complete set of functions that is used to handle the processing between Matlab and C# environment.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MLApp;
using System.Threading;

namespace Irfan.Matlab
{
    public partial class Matlab : Form
    {
        public Matlab()
        {
            InitializeComponent();

        }

        #region Variable and Matlab Functions

        // ------------------------------------------ Variable Region -------------------
--------------//

        public string Concepts;
        public static string ConExt;


        // ------------------------------------------ Matlab Region --------------------
------------//
```

```
        public void uSemanticIntensity()
        {
            // Calling Matlab function
            MLAppClass SE = new MLAppClass();
            Matlab.ConExt = SE.Execute("uSemanticIntensity('" + tbHI.Text + "','" +
tbHA.Text + "','" + tbTrgAnn.Text + "')");
        }
        public void SemanticIntensity()
        {
            MLAppClass SE = new MLAppClass();
            Matlab.ConExt = SE.Execute("SemanticIntensity('" + tbHI.Text + "','" +
tbHA.Text + "','" + tbTrgAnn.Text + "')");
        }
        public void SemanticDB()
        {
            // Setting paths for images and Annotations
            path();
            MLAppClass SE = new MLAppClass();
            Matlab.ConExt = SE.Execute("SemanticDB('" + Concepts + "')");
        }
        public void Database()
        {
            // Setting paths for images and Annotations
            path();
            MLAppClass DB = new MLAppClass();
            Matlab.ConExt = DB.Execute("DBCreation");
            Matlab.ConExt = Matlab.ConExt.Substring(8);
        }
        public void path()
        {
            // Define Directories Path
            MLAppClass IPath = new MLAppClass();
            MLAppClass APath = new MLAppClass();
            IPath.Execute("setImagePath('"+ tbHI.Text + "')");
            APath.Execute("setAnnotationPath('" + tbHA.Text + "')");
        }


        #endregion

        #region Othere btn events

        private void btnHIpath_Click(object sender, EventArgs e)
        {
            tbHI.Text = "D:\\LabelMeDB\\Images";
        }

        private void btnHApath_Click(object sender, EventArgs e)
        {
            tbHA.Text = "D:\\LabelMeDB\\Annotations";
        }

        private void btnConceptListSD_Click(object sender, EventArgs e)
        {
            dataGridView1.DataSource = Main.GridData(Convert.ToDouble(tbThr.Text));
        }

        private void btnResult_Click(object sender, EventArgs e)
```

```
        {
            tbReport.Text = "";
            // Setting paths for images and Annotations
            path();

            // Calling Matlab function
            MLAppClass SE = new MLAppClass();
            int a = Convert.ToInt32(tbRangeres1.Text), b =
Convert.ToInt32(tbRangeres2.Text);
           // string matfun = "resultDisplay2(" + a + "," + b + ")";
           // string conExt = SE.Execute("resultDisplay2(" + a + "," + b + ")");
            tbReport.Text = SE.Execute("resultDisplay2(" + a + "," + b + ")");
        }

        private void button2_Click(object sender, EventArgs e)
        {
            DataTable Pir = new DataTable("ConceptList");
            DataColumn Concept = new DataColumn("Concept");
            DataColumn SS = new DataColumn("SS");
            Pir.Columns.Add(Concept);
            Pir.Columns.Add(SS);
            DataRow newRow;


            for(int i = 0; i<= Main.ConceptSS.Length-1; i++)
            {
               newRow = Pir.NewRow();
                newRow["Concept"] = Main.ConceptSS[i].Concept;
                newRow["SS"] = Main.ConceptSS[i].SS;
                Pir.Rows.Add(newRow);
            }
            dataGridView1.DataSource = Pir;
            MessageBox.Show("Done");
        }

        private void button3_Click(object sender, EventArgs e)
        {
            FolderBrowserDialog fd = new FolderBrowserDialog();
            fd.ShowDialog();
            tbHI.Text = fd.SelectedPath.ToString();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            FolderBrowserDialog fd = new FolderBrowserDialog();
            fd.ShowDialog();
            tbHA.Text = fd.SelectedPath.ToString();
        }

        #endregion

        #region Threads define

        private void button1_Click(object sender, EventArgs e)
        {
            Thread uSemInt = new Thread(uSemanticIntensity);
            uSemInt.Start();
            tbReport.Text = Matlab.ConExt;
```

```
        }

        private void btnSemIntensity_Click(object sender, EventArgs e)
        {
            Thread SI = new Thread(SemanticIntensity);
            SI.Start();
            tbReport.Text = Matlab.ConExt;
        }

        private void btnSemanticExtraction_Click(object sender, EventArgs e)
        {
            Concepts ="";
            for (int i = 0; i <= Main.ConceptSS.Length - 1; i++)
            {
                if (Main.ConceptSS[i].SS>= Convert.ToDouble(tbThr.Text))
                    Concepts += Main.ConceptSS[i].Concept + ',';
            }
            Concepts = Concepts.Substring(0, Concepts.Length - 2);
            Thread SDB = new Thread(SemanticDB);
            SDB.Start();
            tbReport.Text = Matlab.ConExt;
        }

        private void DBCreation_Click(object sender, EventArgs e)
        {
            Thread DB = new Thread(Database);
            DB.Start();
            tbReport.Text = Matlab.ConExt;
        }

        #endregion

        private void button5_Click(object sender, EventArgs e)
        {
            FolderBrowserDialog fd = new FolderBrowserDialog();
            fd.ShowDialog();
            tbTrgAnn.Text = fd.SelectedPath.ToString();
        }
    }
}
```

References

Lawrence et al 2004 Lawrence A. Rowe and Ramesh Jain. Acm sigmm retreat report on future directions in multimedia research. In Proceedings of ACM Multimedia, March 2004.

Ribeiro et al 2001 Ribeiro C., David G.: A Metadata Model for Multimedia Databases, Proceedings of the International Cultural Heritage Informatics Meeting (ichim), Milan, Italy, 2001

Santini et al 1998 S. Santini and R. Jain. Beyond query by example. In MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia, pages 345{350, Bristol, United Kingdom, September 1998.

Smeulders et al 2000 A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12):1349{1380, December 2000.

Datta et al 2008 R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. In ACM Computing Surveys, 2008.

Iskandar 2008 Iskandar 2008 Iskandar D.N.A, "Image Retrieval using Automatic Region Tagging", PhD Thesis, March 2008

Chang et al 1992 S.-K. Chang and A. Hsu. Image information systems: Where do we go from here? Knowledge and Data Engineering, IEEE Transactions on, 4(5):431–442, 1992.

Arnold et al 2000 Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on, 22(12):1349–1380, 2000.

Arun, 2004 Arun, S. (2004). "Metadata management: past, present and future." Decision Support Systems 37(1): 151

Milstead et al 1999 Milstead J., Feldman S.: Metadata: Cataloging by Any Other Name ..., ONLINE Magazine 23(1), Information Today, Medford, USA, January/February 1999

Jain, 1994 Jain R.: Semantics in Multimedia Systems, IEEE Multimedia, 1(2), 1994

Yohan et al (2009) Yohan Jin, Latifur Khan, B. Prabhakaran, "Knowledge Based Image Annotation Refinement", journal of signal processing systems Volume 58, Number 3, 387-406, 2009

G. Shafer (1976). G. Shafer (1976). “A mathematical theory of evidence”. Princeton University Press

Amjad et al (2009) Amjad Altadmri and Amr Ahmed “Video databases annotation enhancing using commonsense knowledge bases for indexing and retrieval”. In: The 13th IASTED International Conference on Artificial Intelligence and Soft Computing. September, 2009, Spain.

Zhu et al (2002) Xingquan Zhu, Jianping Fan, Xiangyang Xue, Lide Wu, Ahmed K. Elmagarmid, "Semi-Automatic Video Content Annotation", Advances In Multimedia Information Processing, Volume 2532, 37-52, 2002

Yan et al (2006) Yan SONG, Xian-Sheng HUA, Guo-Jun Qi, Li-Rong Dai, Meng Wang, Hong-Jiang Zhang, "Efficient semantic annotation method for indexing large personal video database", In the proceedings of the 8th ACM international workshop on Multimedia information retrieval, 2006

Yan et al (2005) Yan SONG, Xian-Sheng HUA, Li-Rong DAI, Ren-Hua WANG, "Semi-Automatic Video Semantic Annotation Based on Active Learning", Visual Communications and Image Processing, Proceedings of SPIE Volume: 5960, 2005

Fischer, (2008) M. Fischer, “Automatic identification of persons in TV series” Universität Karlsruhe (TH) M.S. Thesis, 2008.

Arun, S. (2004). Arun, S. (2004). "Metadata management: past, present and future." Decision Support Systems 37(1): 151

Milstead et al (1999) Milstead J., Feldman S.: Metadata: Cataloging by Any Other Name ..., ONLINE Magazine 23(1), Information Today, Medford, USA, January/February 1999

Turner et al (2002) Turner, Tom: What is metadata?, Kaleidoscope 10(7), Cornell University Library, USA, February 2002

Berners-Lee et al (2001) Berners-Lee T., Hendler J., Lassila O.: The Semantic Web, Scientific American, May 2001

Jain et al (1994) Jain R.: Semantics in Multimedia Systems, IEEE Multimedia, 1(2), 1994

Mojsilovic et al (2001)	Mojsilovic A., Rogowitz B.: Capturing image semantics with low-level descriptors, Proceedings of the International Conference on Image Processing, ICIP 2001, Thessaloniki, Greece, September 2001
Mojsilovic et al (2002)	Mojsilovic A., Gomes J., Rogowitz B.: ISee: Perceptual features for image library navigation, Proceedings of the 2002 SPIE Human Vision and Electronic Imaging conference, San Jose, USA, 2002
Vailaya et al (2001)	Vailaya A., Figueiredo M.A.T., Jain A.K., Zhang H.-J.: Image Classification for Content-Based Indexing, IEEE Transactions on Image Processing, 10(1), January 2001
Lindley C.A. et al (1998)	Lindley C.A., Srinivasan U.: Query Semantics for Content-Based Retrieval of Video Data: An Empirical Investigation, Storage and Retrieval Issues in Image- and Multimedia Databases, August 24-28, in conjunction with 9th International Conference DEXA98, Vienna, Austria, 1998
Zhou X.S. et al (2000)	Zhou X.S., Huang T.S.: CBIR: from Low-Level Features to High-Level Semantics, Proceedings of SPIE Image and Video Communication and Processing 2000, San Jose, USA, January 2000
Martinez A.M. et al (2000)	Martinez A.M., Serra J.R.: A New Approach to Object-related Image Retrieval, Journal of Visual Languages and Computing, 11(3), Academic Press, June 2000
Page K., et al (2001)	Page K., Juby B., Beales R., De Roure D.: Continuous Metadata, Proceedings of the 2nd Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking & Broadcasting, 2001
Weibel S., et al (2001)	Weibel S., Lagoze C.: An element set to support resource discovery, International Journal on Digital Libraries, Volume 1, Number 2, September 1997
Hillman D. (2001)	Hillman D.: Using Dublin Core, WWW-address: http://dublincore.org/documents/usageguide , April 2001
DCMI, (2001)	DCMI: Dublin Core Metadata Initiative (DCMI) Overview, WWW-address: http://dublincore.org/about/overview, 2001
Bray T. et al (2000)	Bray T., Paoli J., Sperberg-McQueen C.-M., Maler E. (eds): Extensible Markup Language (XML) 1.0 (Second Edition), WWW-address: http://www.w3.org/TR/REC-xml, October 2000
Geroimenko V. et al (2002)	Geroimenko V., Chen C: The XML Revolution and the Semantic Web, Visualizing the Semantic Web, Springer, Germany, 2002

Ferraiolo J., et al (2003)	Ferraiolo J., Fujisawa J. Jackson D.: Scalable Vector Graphics (SVG) 1.1 Specification, WWW-address: http://www.w3.org/TR/SVG11/, January 2003
Web 3D consortium (2003)	Web 3D consortium: Extensible 3D (X3D) encodings ISO/IEC 19776-1:200x Part 1 XML encoding, WWW-address: http://www.web3d.org/technicalinfo/specifications/ISO_IEC_19776/Part01/index.html, January 2003
Ayers J. et al (2001)	Ayers J. et al.: Synchronized Multimedia Integration Language (SMIL 2.0), WWW-address: http://www.w3.org/TR/smil20/, August 2001
Dornfest R., et al (2001)	Dornfest R., Brickley D.: The Power of Metadata, WWW-address: http://www.openp2p.com/lpt/a/554, excerpt from: Peer-to-Peer Harnessing the Power of Disruptive Technologies, O'Reilly, USA, 2001
Ianella R. et al (1998)	Ianella R.: An Idiot's guide to the Resource Description Framework, The New Review of Information Networking, 4, 1998
Berners-Lee T., et al (2001)	Berners-Lee T., Hendler J., Lassila O.: The Semantic Web, Scientific American, May 2001
Berners-Lee T., et al (1998)	Berners-Lee T., Fielding R., Irvine U.C., Masinter L.: Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396, WWW-address: http://www.ietf.org/rfc/rfc2396.txt, August 1998
Champin P.-A et al (2001)	Champin P.-A.: RDF Tutorial, WWW-address: http://www710.univ-lyon1.fr/~champin/rdf-tutorial/, April 2001
Nic M. et al (2010)	Nic M.: RDF Tutorial - Part I: basic syntax and containers, WWW-address: http://www.zvon.org/xxl/RDFTutorial/General/book.html, December, 2010
Brickley D., et al (2004)	Brickley D., Guha R.V., McBride B.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft, WWW-address: http://www.w3.org/TR/rdf-schema/, 10 February 2004
Conolly D., et al (2001)	Conolly D., van Harmelen F., Horrocks I., McGuiness D.L., Patel-Schneider P.F., Stein L.A.: DAML+OIL (March 2001) Reference Description, W3C Note, WWW-address: http://www.w3.org/TR/daml+oil-reference, 18 December 2001
Day N., et al (2002)	Day N., Martinez J.M.: Introduction to MPEG-7 (v4.0), ISO/IEC JTC1/SC29/WG11 N4675, Jeju, 2002

Martinez (2002) Martinez J.M.: MPEG-7 Overview (version 8), ISO/IEC JTC1/SC29/WG11N4980, Klangenfurt, 2002

Nack F.-M. et al (2002) Nack F.-M., Hardman H.-L.: Towards a syntax for multimedia semantics, Report INS-R0204, Centrum voor Wiskunde en Informatica, Amsterdam, Netherlands, April 2002

Duygulu, et al (2002) Pinar Duygulu, Kobus Barnard, Joo F. G. de Freitas, and David A. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In Proceedings of European Conference on Computer Vision (ECCV), pages 97–112, 2002.

Mori, et al (1999) Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In First International Workshop on Multimedia Intelligent Storage and Retrieval Management, 1999.

Jeon, et al (2003) J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In SIGIR 03': Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pages 119–126, 2003.

Lavrenko, et al (2003) V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems, volume 16, pages 553–560, 2003.

Monay, et al (2003) F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In Proceedings of the eleventh ACM international conference on Multimedia, pages 275–278, 2003.

Hofmann. (1999) Thomas Hofmann. Probabilistic latent semantic indexing. In Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval, pages 50–57, Berkeley, California, August 1999.

Deerwester, et al (1990) Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41:391–407, 1990.

Blei, et al (2003) David M. Blei and Michael I. Jordan. Modeling annotated data. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 127 – 134, Toronto, Canada, 2003.

Carbonetto, et al (2004) Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In 8th European Conference on Computer Vision (ECCV), pages 350–362, 2004.

Wenyin et al (2001) L. Wenyin, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski & B. Field. Semi-automatic image annotation. Proc. of Interact 2001: Conference on Human-Computer Interaction, pages 326–333, 2001.

Lu et al (2000) Y. Lu, C. Hu, X. Zhu, H.J. Zhang & Q. Yang. A unified framework for semantics and feature based relevance feedback in image retrieval systems. Proceedings of the eighth ACM international conference on Multimedia, pages 31–37, 2000.

Song et al (2005) Yan Song, Xian-Sheng Hua, Li-Rong Dai & MengWang. Semi-automatic video annotation based on active learning with multiple complementary predictors. In MIR ’05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, pages 97–104, New York, NY, USA, 2005. ACM Press.

Ivan et al (2010) Ivan Ivanov, Peter Vajda, Lutz Goldmann, Jong-Seok Lee, Touradj Ebrahimi, "Object-based Tag Propagation for Semi-Automatic Annotation of Images", MIR’10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

Ahn et al (2004) L. von Ahn and L. Dabbish. Labeling images with a computer game. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004), pages 319–326, 2004.

Ahn et al (2006) L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2006), pages 55–64, 2006.

Qi, G.-J et al (2008) Qi, G.-J., Tang, J., Wang, M., Hua, X.-S., Rui, Y., Mei, T., and Zhang, H.-J. 2008. Correlative multilabel video annotation with temporal kernels. ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 5, No. 1, Article 3

Gao, et al (2006) Y. Gao, J. Fan, H. Luo, X. Xue, & R. Jain (2006). Automatic image annotation by incorporating feature hierarchy and boosting to scale up SVM classifiers. In Proceedings of the 14th Annual ACM International Conference on Multimedia (Santa Barbara, CA, USA, October 23–27).

Cusano, et al (2004) Cusano, C., Ciocca, G., & Schettini, R. (2004). Image annotation using SVM. In Proceedings of internet imaging IV, Vol. SPIE 5304.

Yang, et al (2006) C. Yang, M. Dong, & J. Hua (2006) Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 17–22.

Carneiro, et al (2005a) Carneiro, G., & Vasconcelos, N. (2005). Formulating semantic image annotation s a supervised learning problem. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05’).

Carneiro, et al (2005b) Carneiro, G., & Vasconcelos, N. (2005). A database centric view of semantic image annotation and retrieval. In Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in information Retrieval. Salvador, Brazil, 2005.

Chang, et al (2003) Chang, E. Kingshy, G. Sychay, G. & Wu, G. (2003). CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines. IEEE Trans on CSVT, 13(1), 26–28.

Li, et al (2003) Li, J., & Wang, J. Z. Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Transaction on Pattern Analysis and Machine Intelligence, 25(9), 1075–1088. (2003)

Wei-Chao, et al (2010) Wei-Chao Lin , MichaelOakes, JohnTait "Improving image annotation via representative feature vector selection", Neurocomputing , Volume 73, Issues 10-12, June 2010, Pages 1774-1782

Pan, et al (2004) Pan, J. Y., Yang, H. J., Faloutsos, C., & Duygulu, P. (2004). Automatic multimedia cross-modal correlation discovery. In Proceedings of the 10th ACM SIGKDD Conference KDD 2004. Seattle, WA, 653–658.

Kang, et al (2004) Kang, F., Jin, R., & Chai, J. Y. (2004). Regularizing translation models for better automatic image annotation. In Proceedings of The Thirteenth Conference on Information and Knowledge Management, 2004, Washington D. C., USA, Nov. 8-13, 350-359.

Guangyu Zhu et al.(2010) Guangyu Zhu, Shuicheng Yan, Yi Ma "Image Tag Refinement Towards Low-Rank, Content-Tag Prior and Error Sparsity", ACM multimedia 2010 International conference, October 25–29, 2010, Firenze, Italy.

Lavrenko, et al.(2004) Lavrenko, V. Feng, S. L., & Manmatha (2004). Statistical models for automatic video annotation and retrieval. International Conference on Acoustics, Speech and Signal Processing, (ICASSP) Montreal, QC, Canada, 17–21.

Barrat, et al.(2010) Sabine Barrat, Salvatore Tabbone, "Modeling, classifying and annotating weakly annotated images using Bayesian network", Journal of Visual Communication and Image Representation Volume 21, Issue 4, May 2010, Pages 355-363

Yohan, et al.(2005) Yohan, J., Khan, L., Wang, L., & Awad, M. (2005) Image annotations by combining multiple evidence and WordNet. In Proceedings the 13th annual ACM international conference on Multimedia (MM05'), Singapore, 706–715.

Hirst, et al. (1986) Hirst,G. and St-Onge, D. Lexical chains as representations of context for the detection and correction of malapropisms. In WordNet, An Electronic Lexical Database. The MIT Press, 1986.

Furnas, et al. (1987) G. W. Furnas, T. K. Landauer, L. M. Gomez,, and S. T. Dumais, "The Vocabulary Problem in Human-System Communication", Communications of the ACM, Vol. 30, No. 11, November 1987, 964-971.

Covington, et al. (2007) M.A. Covington, J. D. McFall,"Using MontyLingua 2.1 with C# and Microsoft.Net", CASPR Research Report, 2007

Carneiro, et al. (2005)	Carneiro, G., & Vasconcelos, N. (2005). Formulating semantic image annotation s a supervised learning problem. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05').
Liu, et al. (2004)	H. Liu and P. Singh. ConceptNet a practical commonsense reasoning tool-kit. BT Technology Journal, 2(4):211-226, 2004.
Hsu, et al. (2008)	M. H. Hsu, M. F. Tsai, and H. H. Chen. Combining wordnet and conceptnet for automatic query expansion: a learning approach. In Asia Information Retrieval Symposium, volume 4993, pages 213-224. Springer, 2008.
Leacock, et al. (1998)	Leacock,C. and Chodorow,M. Combining local context and WordNet sense similarity for word sense identification. InWordNet, An Electronic Lexical Database. The MIT Press, 1998
Lesk, et al. (1986)	Lesk,M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the SIGDOC Conference, 1986.
Wu, et al. (1994)	Wu,Z. and Palmer,M. Verb semantics and lexical selection. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1994.
Resnik, et al. (1995)	Resnik,P. Using information content to evaluate semantic similarity. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995.
Lin, et al. (1998)	Lin,D. An Information-theoretic Definition of Similarity. In Proc. of the ICML'98, 1998.
Jiang, et al. (1997)	Jiang,J. and Conrath,D. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of the International Conference on Research in Computational Linguistics, 1997.
Lenat, et al. (1995)	D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," Communications of the ACM, vol. 38, no. 11, pp. 33–38, 1995.
Fellbaum, et al. (1998)	C. Fellbaum, WordNet: an electronic lexical database. Cambridge, Mass: MIT Press, 1998.
Fei-Fei, et al (2007)	Fei-Fei, L., Fergus, R., & Perona, P. (2007, in press). One-shot learning of object categories. IEEE Transactions on Pattern Recognition and Machine Intelligence. The Caltech 101 dataset can be downloaded at http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html.

Winn, et al (2005) Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In IEEE international conference on computer vision. The MSRC dataset can be downloaded at http://research.microsoft.com/en-us/projects/objectclassrecognition/

Bileschi, et al (2006) Bileschi, S. (2006). CBCL streetscenes (Technical report).MIT CBCL. The CBCL-Streetscenes dataset can be downloaded at http://cbcl.mit.edu/software-datasets.

Everingham, et al (2006) Everingham, M., Zisserman, A., Williams, C. K. I., & Van Gool, L. (2006). The pascal visual object classes challenge 2006 (voc 2006) results (Technical report). September 2006. The PASCAL2006 dataset can be downloaded at http://www.pascal-etwork.org/challenges/VOC/voc2006/

Kang, et al (2006) F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. CVPR, pages 291–294, 1719-1726 2006.

Naphade, et al (2006) M. Naphade, J. R. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, and A. Hauptmann. Large-scale concept ontology for multimedia. IEEE Multimedia, 13:86–91, Jul-Sep 2006.

Szummer, et al (1998) M. Szummer and R. Picard. Indoor-outdoor image classification. Workshop Content-Based Access to Image and Video Databases, 1998.

Haering, et al (1997) Z. M. N. Haering and N. Lobo. Locating dedicuous trees. Proc. Workshop in Content-Based Access to
Image and Video Libraries, pages 18–25, 1997.

Vailaya, et al (1998) A. J. A. Vailaya and H. Zhang. On image classification: City vs. landscape. Pattern Recognition, pages 1921–1936, 1998.

Forsyth, et al (1997) D. Forsyth and M. Fleck. Body plans. Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pages 678–683, 1997.

Li, et al (2002) Y. Li and L. Shapiro. Consistent line clusters for building recognition in CBIR. Proc. International Conf. Pattern Recognition, pages 952–956, 2002.

Duygulu, et al (2002) P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. ECCV, 2002.

Gao, et al (2006) Y. Gao, J. Fan, X. Xue, and R. Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. ACM Multimedia, pages 901–910, 2006.

Feng, et al (2004) S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. CVPR, pages 1002–1009, 2004.

Liu, et al (2007) J. Liu, B. Wang, M. Li, Z. Li, W. Ma, H. Lu, and S. Ma. Dual cross-media relevance model for image annotation. ACM Multimedia, pages 605–614, 2007.

Jin, et al (2004) R. Jin, J. Y. Chai, and L. Si. Effective automatic image annotation via a coherent language model and active learning. ACM Multimedia, pages 892–899, 2004.

Jin, et al (2005) Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & WordNet. ACM Multimedia, pages 706–715, 2005.

Shi, et al (2006) R. Shi, T. Chua, C. lee, and S. Gao. Bayesian learning of hierarchical multinomial mixture models of concepts for automatic image annotation. CIVR, pages 102–112, 2006.

Zhou, et al (2007) X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi. Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. CIVR, pages 25–32, 2007.

Qi, et al (2007) G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang. Correlative multi-label video annotation. ACM Multimedia, pages 17–26, 2007.

Amaral, et al (2010) Igor F. Amaral, Filipe Coelho, Joaquim F. Pinto da Costa and Jaime S. Cardoso, "Hierarchical Medical Image Annotation Using SVM-based Approaches", IEEE, 2010

Frakes, et al 1998 Frakes, William B. and Baeza-Yates, Ricardo (Eds.), Information Retrieval: Data Structures and Algorithms, Englewood Cliffs, NJ: Prentice-Hall, 1992. ISBN: 0-13-463837-9 (504 pgs.) (Revised Version - 1998) republished on a cd-rom entitled Dr.Dobbs Essential Books on Algorithms and Data Structures

[Brunelli et al. 1999] Brunelli, R., Mich, O., & Modena, C. M. (1999). A survey on the automatic indexing of video data. Journal of Visual Communication and Image Representation, 10(2), 78–112.

[Wang et al. 2000] Wang, Y., Liu, Z., & Huang, J-C. (2000). Multimedia content analysis using both audio and visual clues. IEEE Signal Processing, 17(6), 12–36.

[Snoek et al. 2005] Snoek, C. G. M., & Worring, M. (2005). Multimedia event based video indexing using time intervals. IEEE Transactions on Multimedia, 7(4), 638-647.

[Benitez et al. 2002] Benitez, A. B. et al. (2002). Semantics of multimedia in MPEG-7. In Proceedings of the
IEEE International Conference on Image Processing, Rochester, NY.

[Monaco. 2009] James Monaco. How to Read a Film. Oxford Press, London, 4th edition, 2009. ISBN 978-0-19-532105-0.

[Zhang et al. 1997] Hong Jiang Zhang, Jianhua Wu, Di Zhong, and Stephen W. Smoliar. An integrated system for content-based video retrieval and browsing. Pattern Recognition, 30(4): 643–658, 1997.

[Hauptmann et al. 2007] A. G. Hauptmann, M. Y. Chen, M. Christel, W. H. Lin, and J. Yang, "A hybrid approach to improving semantic extraction of news video," in International Conference on Semantic Computing, 2007. ICSC 2007., 2007, pp. 79–86.

[Bagdanov et al. 2007] A. D. Bagdanov, M. Bertini, A. D. Bimbo, G. Serra, and C. Torniai, "Semantic annotation and retrieval of video events using multimedia ontologies," in International Conference on Semantic Computing, 2007, pp. 713–720.

[Yuan et al. 2008] P. Yuan, B. Zhang, and J. Li, "Semantic concept learning through massive internet video mining," in IEEE International Conference on Data Mining Workshops, 2008, pp. 847–853.

[Felbaum. 1998] C. Fellbaum, WordNet: an electronic lexical database. Cambridge, Mass: MIT Press, 1998.

[Shevade et al. 2006] B. Shevade and H. Sundaram, "A visual annotation framework using common-sensical and linguistic relationships for semantic media retrieval," LECTURE NOTES IN COMPUTER SCIENCE, vol. 3877, p. 251, 2006.

[Girgensohn et al., 2005] A. Girgensohn, J. Adcock, M. Cooper, and L. Wilcox. A Synergistic Approach to Efficient Interactive Video Retrieval. In Proc. Human-Computer Interaction INTERACT 2005, LNCS 3585, pages 781–794. Technische Hogeschool Eindhoven, The Netherlands, 2005.

MPEG-4, 1996 "Description of MPEG-4", ISO/IEC JTC1/SC29/WG11 N1410, MPEG document N1410 Oct. 1996.

MPEG-7, 2000 67. "Introduction to MPEG-7 (version 1.0)", ISO/IEC JTC1/SC29/WG11 N3545, Beijing, July 2000

[Yeung et al. 1996] M.M. Yeung, B.L. Yeo and B. Liu, "Extracting Story Units from Long Programs for Video Browsing and Navigation", International Conference on Multimedia Computing and Systems, June 1996.

[Zhang et al. 1994] HongJiang Zhang, Yihong Gong, etc. "Automatic Parsing of News Video", Proc. IEEE Int'l Conf. Multimedia Computing and Systems, IEEE Computer Society Press, Los Alamitos, Calif.,1994.

[Zhong et al. 1996] Di Zhong, H. J. Zhang and S.-F.Chang, "Clustering methods for video browsing and annotation", Storage & Retrieval for Still Image and Video Databases IV, IS&T/SPIE's Electronic Imaging: Science & Technology, Feb. 96.

[Bimbo et al. 1995] A.D. Bimbo, E. Vicario and D. Zingoni, "Symbolic description and visual querying of image sequences using spatio-temporal logic", IEEE Transactions on Knowledge and Data Engineering, Vol 7, No. 4, August, 1995.

[Chang et al. 1987] S.-K. Chang, Q. Y. Shi, and C. Y. Yan, "Iconic indexing by 2-D strings", IEEE Trans. Pattern Anal. Machine Intell., 9(3):413-428, May 1987.

[Wang et al. 1996] H. Wang and S.-F. Chang, "Automatic Face Region Detection in MPEG Video Sequences", SPIE's Photonics China '96 - Electronic Imaging and Multimedia Systems, Beijing, China, November 1996.

[Zhang et al. 1995] H. J. Zhang et al., "Automatic Parsing and Indexing of News Video", Multimedia Systems, 2 (6), pp. 256-266, 1995.

[Meng et al. 1996] J. Meng and S.-F. Chang, "Tools for Compressed-Domain Video Indexing and Editing", SPIE Conference on Storage and Retrieval for Image and Video Database, San Jose, Feb. 1996.

[Chandrasekaran et al. 1999] B. Chandrasekaran, J. Josephson, and V. Benjamins, "What are ontologies, and why do we need them?" IEEE Intelligent Systems and their applications, vol. 14, no. 1, pp. 20–26, 1999.

[Felbaum.1998] C. Fellbaum, WordNet: an electronic lexical database. Cambridge, Mass: MIT Press, 1998.

[Lenat et al. 1995] D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," Communications of the ACM, vol. 38, no. 11, pp. 33–38, 1995.

[Liu et al. 2004] H. Liu and P. Singh, "Conceptnet a practical commonsense reasoning tool-kit," BT Technology Journal, vol. 22, no. 4, pp. 211–226, 2004.

[Naphade et al. 2002] M. Naphade, I. Kozintsev, and T. Huang. Factor graph framework for semantic video indexing. IEEE Trans. on CSVT, 12(1), Jan. 2002.

[Snoek et al. 2006] C. Snoek and et al. The challenge problem for automated detection of 101 semantic concepts in multimedia. In Proceedings of the ACM International Conference on Multimedia, pages 421–430, Santa Barbara, USA, October 2006.

[Naphade et al. 2005] M. R. Naphade, L. Kennedy, J. R. Kender, S.-F. Chang, J. R. Smith, P. Over, and A. Hauptmann. A light scale concept ontology for multimedia understanding for TRECVID 2005. In IBM Research Report RC23612 (W0505-104), 2005.

[Smeaton et al. 2006] A. F. Smeaton and P. Wilkins. TRECVID 2004: Interactive Search Questionnaires. http://www-nlpir.nist.gov/projects/tv2004/questionnaires.html, 09 2004. last checked: 26.07.2006.

Meng, et al. 1996 J. Meng and S.-F. Chang, "Tools for Compressed-Domain Video Indexing and Editing", SPIE Conference on Storage and Retrieval for Image and Video Database, San Jose, Feb. 1996.

Zhang, et al., et al. 1995 H. J. Zhang et al., "Automatic Parsing and Indexing of News Video", Multimedia Systems, 2 (6), pp. 256-266, 1995.

Wang, et al., et al. 1996 H. Wang and S.-F. Chang, "Automatic Face Region Detection in MPEG Video Sequences", SPIE's Photonics China '96 - Electronic Imaging and Multimedia Systems, Beijing, China, November 1996.

Chang, et al., et al. 1987 S.-K. Chang, Q. Y. Shi, and C. Y. Yan, "Iconic indexing by 2-D strings", IEEE Trans. Pattern Anal. Machine Intell., 9(3):413-428, May 1987.

Bimbo, et al., et al. 1995 A.D. Bimbo, E. Vicario and D. Zingoni, "Symbolic description and visual querying of image sequences using spatio-temporal logic", IEEE Transactions on Knowledge and Data Engineering, Vol 7, No. 4, August, 1995.

Zhong, et al., et al. 1996 Di Zhong, H. J. Zhang and S.-F.Chang, "Clustering methods for video browsing and annotation", Storage & Retrieval for Still Image and Video Databases IV, IS&T/SPIE's Electronic Imaging: Science & Technology, Feb. 96.

Zhang, et al., et al. 1994 Hong Jiang Zhang, Yihong Gong, etc. "Automatic Parsing of News Video", Proc. IEEE Int'l Conf. Multimedia Computing and Systems, IEEE Computer Society Press, Los Alamitos, Calif.,1994.

Wang, et al., et al. 2007 James Ze Wang, Nozha Boujemaa, Alberto Del Bimbo, and Jia Li, editors. MIR'07: Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval, Augsburg, Bavaria, Germany, 2007a. ACM. ISBN 978-1-59593- 778-0.

Yohan et al (2009) Yohan Jin, Latifur Khan, B. Prabhakaran, "Knowledge Based Image Annotation Refinement", journal of signal processing systems Volume 58, Number 3, 387-406, 2009

G. Shafer (1976).	G. Shafer (1976). “A mathematical theory of evidence”. Princeton University Press
Amjad et al (2009)	Amjad Altadmri and Amr Ahmed “Video databases annotation enhancing using commonsense knowledge bases for indexing and retrieval”. In: The 13th IASTED International Conference on Artificial Intelligence and Soft Computing. September, 2009, Spain.
Xingquan Zhu et al (2002)	Xingquan Zhu, Jianping Fan, Xiangyang Xue, Lide Wu, Ahmed K. Elmagarmid, "Semi-Automatic Video Content Annotation", Advances In Multimedia Information Processing, Volume 2532, 37-52, 2002
Yan SONG et al (2006)	Yan SONG, Xian-Sheng HUA, Guo-Jun Qi, Li-Rong Dai, Meng Wang, Hong-Jiang Zhang, "Efficient semantic annotation method for indexing large personal video database", In the proceedings of the 8th ACM international workshop on Multimedia information retrieval, 2006
Yan SONG et al (2005)	Yan SONG, Xian-Sheng HUA, Li-Rong DAI, Ren-Hua WANG, "Semi-Automatic Video Semantic Annotation Based on Active Learning", Visual Communications and Image Processing, Proceedings of SPIE Volume: 5960, 2005
M. Fischer, (2008)	M. Fischer, “Automatic identification of persons in TV series” Universität Karlsruhe (TH) M.S. Thesis, 2008.
Arun, S. (2004).	Arun, S. (2004). "Metadata management: past, present and future." Decision Support Systems 37(1): 151
Milstead et al (1999)	Milstead J., Feldman S.: Metadata: Cataloging by Any Other Name ..., ONLINE Magazine 23(1), Information Today, Medford, USA, January/February 1999
Turner et al (2002)	Turner, Tom: What is metadata?, Kaleidoscope 10(7), Cornell University Library, USA, February 2002
Berners-Lee et al (2001)	Berners-Lee T., Hendler J., Lassila O.: The Semantic Web, Scientific American, May 2001
Jain et al (1994)	Jain R.: Semantics in Multimedia Systems, IEEE Multimedia, 1(2), 1994
Mojsilovic et al (2001)	Mojsilovic A., Rogowitz B.: Capturing image semantics with low-level descriptors, Proceedings of the International Conference on Image Processing, ICIP 2001, Thessaloniki, Greece, September 2001

Mojsilovic et al (2002)	Mojsilovic A., Gomes J., Rogowitz B.: ISee: Perceptual features for image library navigation, Proceedings of the 2002 SPIE Human Vision and Electronic Imaging conference, San Jose, USA, 2002
Vailaya et al (2001)	Vailaya A., Figueiredo M.A.T., Jain A.K., Zhang H.-J.: Image Classification for Content-Based Indexing, IEEE Transactions on Image Processing, 10(1), January 2001
Lindley C.A. et al (1998)	Lindley C.A., Srinivasan U.: Query Semantics for Content-Based Retrieval of Video Data: An Empirical Investigation, Storage and Retrieval Issues in Image- and Multimedia Databases, August 24-28, in conjunction with 9th International Conference DEXA98, Vienna, Austria, 1998
Zhou X.S. et al (2000)	Zhou X.S., Huang T.S.: CBIR: from Low-Level Features to High-Level Semantics, Proceedings of SPIE Image and Video Communication and Processing 2000, San Jose, USA, January 2000
Martinez A.M. et al (2000)	Martinez A.M., Serra J.R.: A New Approach to Object-related Image Retrieval, Journal of Visual Languages and Computing, 11(3), Academic Press, June 2000
Page K., et al (2001)	Page K., Juby B., Beales R., De Roure D.: Continuous Metadata, Proceedings of the 2nd Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking & Broadcasting, 2001
Weibel S., et al (2001)	Weibel S., Lagoze C.: An element set to support resource discovery, International Journal on Digital Libraries, Volume 1, Number 2, September 1997
Hillman D. (2001)	Hillman D.: Using Dublin Core, WWW-address: http://dublincore.org/documents/usageguide , April 2001
DCMI, (2001)	DCMI: Dublin Core Metadata Initiative (DCMI) Overview, WWW-address: http://dublincore.org/about/overview, 2001
Bray T. et al (2000)	Bray T., Paoli J., Sperberg-McQueen C.-M., Maler E. (eds): Extensible Markup Language (XML) 1.0 (Second Edition), WWW-address: http://www.w3.org/TR/REC-xml, October 2000
Geroimenko V. et al (2002)	Geroimenko V., Chen C: The XML Revolution and the Semantic Web, Visualizing the Semantic Web, Springer, Germany, 2002
Ferraiolo J., et al (2003)	Ferraiolo J., Fujisawa J. Jackson D.: Scalable Vector Graphics (SVG) 1.1 Specification, WWW-address: http://www.w3.org/TR/SVG11/, January 2003

Web 3D consortium (2003)	Web 3D consortium: Extensible 3D (X3D) encodings ISO/IEC 19776-1:200x Part 1 XML encoding, WWW-address: http://www.web3d.org/technicalinfo/specifications/ISO_IEC_19776/Part01/index.html, January 2003
Ayers J. et al (2001)	Ayers J. et al.: Synchronized Multimedia Integration Language (SMIL 2.0), WWW-address: http://www.w3.org/TR/smil20/, August 2001
Dornfest R., et al (2001)	Dornfest R., Brickley D.: The Power of Metadata, WWW-address: http://www.openp2p.com/lpt/a/554, excerpt from: Peer-to-Peer Harnessing the Power of Disruptive Technologies, O'Reilly, USA, 2001
Ianella R. et al (1998)	Ianella R.: An Idiot's guide to the Resource Description Framework, The New Review of Information Networking, 4, 1998
Berners-Lee T., et al (2001)	Berners-Lee T., Hendler J., Lassila O.: The Semantic Web, Scientific American, May 2001
Berners-Lee T., et al (1998)	Berners-Lee T., Fielding R., Irvine U.C., Masinter L.: Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396, WWW-address: http://www.ietf.org/rfc/rfc2396.txt, August 1998
Champin P.-A et al (2001)	Champin P.-A.: RDF Tutorial, WWW-address: http://www710.univ-lyon1.fr/~champin/rdf-tutorial/, April 2001
Nic M. et al (2010)	Nic M.: RDF Tutorial - Part I: basic syntax and containers, WWW-address: http://www.zvon.org/xxl/RDFTutorial/General/book.html, December, 2010
Brickley D., et al (2004)	Brickley D., Guha R.V., McBride B.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft, WWW-address: http://www.w3.org/TR/rdf-schema/, 10 February 2004
Conolly D., et al (2001)	Conolly D., van Harmelen F., Horrocks I., McGuiness D.L., Patel-Schneider P.F., Stein L.A.: DAML+OIL (March 2001) Reference Description, W3C Note, WWW-address: http://www.w3.org/TR/daml+oil-reference, 18 December 2001
Day N., et al (2002)	Day N., Martinez J.M.: Introduction to MPEG-7 (v4.0), ISO/IEC JTC1/SC29/WG11 N4675, Jeju, 2002
Martinez (2002)	Martinez J.M.: MPEG-7 Overview (version 8), ISO/IEC JTC1/SC29/WG11N4980, Klangenfurt, 2002

Nack F.-M. et al (2002)	Nack F.-M., Hardman H.-L.: Towards a syntax for multimedia semantics, Report INS-R0204, Centrum voor Wiskunde en Informatica, Amsterdam, Netherlands, April 2002
Duygulu, et al (2002)	Pinar Duygulu, Kobus Barnard, Joo F. G. de Freitas, and David A. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In Proceedings of European Conference on Computer Vision (ECCV), pages 97–112, 2002.
Mori, et al (1999)	Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In First International Workshop on Multimedia Intelligent Storage and Retrieval Management, 1999.
Jeon, et al (2003)	J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In SIGIR 03': Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pages 119–126, 2003.
Lavrenko, et al (2003)	V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems, volume 16, pages 553–560, 2003.
Monay, et al (2003)	F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In Proceedings of the eleventh ACM international conference on Multimedia, pages 275–278, 2003.
Hofmann. (1999)	Thomas Hofmann. Probabilistic latent semantic indexing. In Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval, pages 50–57, Berkeley, California, August 1999.
Deerwester, et al (1990)	Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41:391–407, 1990.
Blei, et al (2003)	David M. Blei and Michael I. Jordan. Modeling annotated data. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 127 – 134, Toronto, Canada, 2003.
Carbonetto, et al (2004)	Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In 8th European Conference on Computer Vision (ECCV), pages 350–362, 2004.
Wenyin et al (2001)	L. Wenyin, S. Dumais, Y. Sun, H. Zhang, M. Czerwinski & B. Field. Semi-automatic image annotation. Proc. of Interact 2001: Conference on Human-Computer Interaction, pages 326–333, 2001.
Lu et al (2000)	Y. Lu, C. Hu, X. Zhu, H.J. Zhang & Q. Yang. A unified framework for semantics and feature based relevance feedback in image retrieval systems. Proceedings of the eighth ACM international conference on Multimedia, pages 31–37, 2000.

Song et al (2005) Yan Song, Xian-Sheng Hua, Li-Rong Dai & MengWang. Semi-automatic video annotation based on active learning with multiple complementary predictors. In MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, pages 97–104, New York, NY, USA, 2005. ACM Press.

Ivan et al (2010) Ivan Ivanov, Peter Vajda, Lutz Goldmann, Jong-Seok Lee, Touradj Ebrahimi, "Object-based Tag Propagation for Semi-Automatic Annotation of Images", MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

Ahn et al (2004) L. von Ahn and L. Dabbish. Labeling images with a computer game. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004), pages 319–326, 2004.

Ahn et al (2006) L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2006), pages 55–64, 2006.

Qi, G.-J et al (2008) Qi, G.-J., Tang, J., Wang, M., Hua, X.-S., Rui, Y., Mei, T., and Zhang, H.-J. 2008. Correlative multilabel video annotation with temporal kernels. ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 5, No. 1, Article 3

Gao, et al (2006) Y. Gao, J. Fan, H. Luo, X. Xue, & R. Jain (2006). Automatic image annotation by incorporating feature hierarchy and boosting to scale up SVM classifiers. In Proceedings of the 14th Annual ACM International Conference on Multimedia (Santa Barbara, CA, USA, October 23–27).

Cusano, et al (2004) Cusano, C., Ciocca, G., & Schettini, R. (2004). Image annotation using SVM. In Proceedings of internet imaging IV, Vol. SPIE 5304.

Yang, et al (2006) C. Yang, M. Dong, & J. Hua (2006) Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 17–22.

Carneiro, et al (2005a) Carneiro, G., & Vasconcelos, N. (2005). Formulating semantic image annotation s a supervised learning problem. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05').

Carneiro, et al (2005b) Carneiro, G., & Vasconcelos, N. (2005). A database centric view of semantic image annotation and retrieval. In Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in information Retrieval. Salvador, Brazil, 2005.

Chang, et al (2003) Chang, E. Kingshy, G. Sychay, G. & Wu, G. (2003). CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines. IEEE Trans on CSVT, 13(1), 26–28.

Li, et al (2003)	Li, J., & Wang, J. Z. Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Transaction on Pattern Analysis and Machine Intelligence, 25(9), 1075–1088. (2003)
Wei-Chao, et al (2010)	Wei-Chao Lin , MichaelOakes, JohnTait "Improving image annotation via representative feature vector selection", Neurocomputing , Volume 73, Issues 10-12, June 2010, Pages 1774-1782
Pan, et al (2004)	Pan, J. Y., Yang, H. J., Faloutsos, C., & Duygulu, P. (2004). Automatic multimedia cross-modal correlation discovery. In Proceedings of the 10th ACM SIGKDD Conference KDD 2004. Seattle, WA, 653–658.
Kang, et al (2004)	Kang, F., Jin, R., & Chai, J. Y. (2004). Regularizing translation models for better automatic image annotation. In Proceedings of The Thirteenth Conference on Information and Knowledge Management, 2004, Washington D. C., USA, Nov. 8-13, 350-359.
Zhu et al.(2010)	Guangyu Zhu, Shuicheng Yan, Yi Ma "Image Tag Refinement Towards Low-Rank, Content-Tag Prior and Error Sparsity", ACM multimedia 2010 International conference, October 25–29, 2010, Firenze, Italy.
Lavrenko, et al.(2004)	Lavrenko, V. Feng, S. L., & Manmatha (2004). Statistical models for automatic video annotation and retrieval. International Conference on Acoustics, Speech and Signal Processing, (ICASSP) Montreal, QC, Canada, 17–21.
Yohan, et al.(2005)	Yohan, J., Khan, L., Wang, L., & Awad, M. (2005) Image annotations by combining multiple evidence and WordNet. In Proceedings the 13th annual ACM international conference on Multimedia (MM05'), Singapore, 706–715.
Hirst, et al. (1986)	Hirst,G. and St-Onge, D. Lexical chains as representations of context for the detection and correction of malapropisms. In WordNet, An Electronic Lexical Database. The MIT Press, 1986.
Furnas, et al. (1987)	G. W. Furnas, T. K. Landauer, L. M. Gomez,, and S. T. Dumais, "The Vocabulary Problem in Human-System Communication", Communications of the ACM, Vol. 30, No. 11, November 1987, 964-971.
Covington, et al. (2007)	M.A. Covington, J. D. McFall,"Using MontyLingua 2.1 with C# and Microsoft.Net", CASPR Research Report, 2007
Carneiro, et al. (2005)	Carneiro, G., & Vasconcelos, N. (2005). Formulating semantic image annotation s a supervised learning problem. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05').
Liu, et al. (2004)	H. Liu and P. Singh. ConceptNet a practical commonsense reasoning tool-kit. BT Technology Journal, 2(4):211-226, 2004.

Hsu, et al. (2008) M. H. Hsu, M. F. Tsai, and H. H. Chen. Combining wordnet and conceptnet for automatic query expansion: a learning approach. In Asia Information Retrieval Symposium, volume 4993, pages 213-224. Springer, 2008.

Leacock, et al. (1998) Leacock,C. and Chodorow,M. Combining local context and WordNet sense similarity for word sense identification. InWordNet, An Electronic Lexical Database. The MIT Press, 1998

Lesk, et al. (1986) Lesk,M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the SIGDOC Conference, 1986.

Wu, et al. (1994) Wu,Z. and Palmer,M. Verb semantics and lexical selection. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1994.

Resnik, et al. (1995) Resnik,P. Using information content to evaluate semantic similarity. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995.

Lin, et al. (1998) Lin,D. An Information-theoretic Definition of Similarity. In Proc. of the ICML'98, 1998.

Jiang, et al. (1997) Jiang,J. and Conrath,D. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of the International Conference on Research in Computational Linguistics, 1997.

Lenat, et al. (1995) D. B. Lenat, “Cyc: A large-scale investment in knowledge infrastructure,” Communications of the ACM, vol. 38, no. 11, pp. 33–38, 1995.

Fellbaum, et al. (1998) C. Fellbaum, WordNet: an electronic lexical database. Cambridge, Mass: MIT Press, 1998.

Fei-Fei, et al (2007) Fei-Fei, L., Fergus, R., & Perona, P. (2007, in press). One-shot learning of object categories. IEEE Transactions on Pattern Recognition and Machine Intelligence. The Caltech 101 dataset can be downloaded at http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html.

Winn, et al (2005) Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In IEEE international conference on computer vision. The MSRC dataset can be downloaded at http://research.microsoft.com/en-us/projects/objectclassrecognition/

Bileschi, et al (2006) Bileschi, S. (2006). CBCL streetscenes (Technical report).MIT CBCL. The CBCL-Streetscenes dataset can be downloaded at http://cbcl.mit.edu/software-datasets.

Everingham, et al (2006) Everingham, M., Zisserman, A., Williams, C. K. I., & Van Gool, L. (2006). The pascal visual object classes challenge
2006 (voc 2006) results (Technical report). September 2006. The PASCAL2006 dataset can be downloaded at http://www.
pascal-network.org/challenges/VOC/voc2006/

Kang, et al (2006) F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning.
CVPR, pages 291–294, 1719-1726 2006.

Naphade, et al (2006) M. Naphade, J. R. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, and A. Hauptmann. Large-scale concept ontology for multimedia. IEEE Multimedia, 13:86–91, Jul-Sep 2006.

Szummer, et al (1998) M. Szummer and R. Picard. Indoor-outdoor image classification. Workshop Content-Based Access to Image and Video Databases, 1998.

Haering, et al (1997) Z. M. N. Haering and N. Lobo. Locating dedicuous trees. Proc. Workshop in Content-Based Access to Image and Video Libraries, pages 18–25, 1997.

Vailaya, et al (1998) A. J. A. Vailaya and H. Zhang. On image classification: City vs. landscape. Pattern Recognition, pages 1921–1936, 1998.

Forsyth, et al (1997) D. Forsyth and M. Fleck. Body plans. Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pages 678–683, 1997.

Li, et al (2002) Y. Li and L. Shapiro. Consistent line clusters for building recognition in CBIR. Proc. International Conf. Pattern Recognition, pages 952–956, 2002.

Duygulu, et al (2002) P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. ECCV, 2002.

Gao, et al (2006) Y. Gao, J. Fan, X. Xue, and R. Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. ACM Multimedia, pages 901–910, 2006.

Feng, et al (2004) S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. CVPR, pages 1002–1009, 2004.

Liu, et al (2007) J. Liu, B. Wang, M. Li, Z. Li, W. Ma, H. Lu, and S. Ma. Dual cross-media relevance model for image annotation. ACM Multimedia, pages 605–614, 2007.

Jin, et al (2004) R. Jin, J. Y. Chai, and L. Si. Effective automatic image annotation via a coherent language model and active learning. ACM Multimedia, pages 892–899, 2004.

Jin, et al (2005) Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & WordNet. ACM Multimedia, pages 706–715, 2005.

Shi, et al (2006) R. Shi, T. Chua, C. lee, and S. Gao. Bayesian learning of hierarchical multinomial mixture models of concepts for automatic image annotation. CIVR, pages 102–112, 2006.

Zhou, et al (2007) X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi. Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. CIVR, pages 25–32, 2007.

Qi, et al (2007) G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang. Correlative multi-label video annotation.
ACM Multimedia, pages 17–26, 2007.

Amaral, et al (2010) Igor F. Amaral, Filipe Coelho, Joaquim F. Pinto da Costa and Jaime S. Cardoso, "Hierarchical Medical Image Annotation Using SVM-based Approaches", IEEE, 2010

Frakes, et al 1998 Frakes, William B. and Baeza-Yates, Ricardo (Eds.), Information Retrieval: Data Structures and Algorithms, Englewood Cliffs, NJ: Prentice-Hall, 1992. ISBN: 0-13-463837-9 (504 pgs.) (Revised Version - 1998) republished on a cd-rom entitled Dr.Dobbs Essential Books on Algorithms and Data Structures

Furnas, et al 1997 Furnas, G. W. (1997), Effective View Navigation, in S. Pemberton (ed.), Proceedings of CHI'97: Human Factors in Computing Systems, ACM Press, pp.367–74.

Witten, et al 1999 Ian H. Witten, Alistar Moffat, Timothy C. Bell, "Managing Gigabytes", Morgan Kaufmann, pages 72115
(Section 3), 1999

Chang, 2002 S.-F. Chang, "The holy grail of content-based media analysis," IEEE Multimedia, vol. 9, no. 2, pp. 6–10, Apr.-Jun. 2002.

Bertini, et al 2004 M. Bertini, A. Del Bimbo, A. Prati, and R. Cucchiara, "Semantic annotation and transcoding for sport videos," in Proc. Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004), Lisboa, Portugal, Apr 2004.

Chang, et al 2001 S.-F. Chang, T. Sikora, and A. Puri, "Overview of the MPEG-7 standard," IEEE Trans. on Circuits and Systems for Video Technology, vol. 11, no. 6, pp. 688–695, June 2001.

Salembier, et al 1999 P. Salembier and F. Marques, “Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services,” IEEE Trans. Circuits and Systems for Video Technology, vol. 9, no. 8, pp. 1147–1169, December 1999.

Al-Khatib, et al 1999 W. Al-Khatib, Y.F. Day, A. Ghafoor, and P.B. Berra, “Semantic modeling and knowledge representation in multimedia databases,” IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 1, pp. 64–80, Jan/Feb 1999.

Yanagawa, et al 2007 A. Yanagawa, S.-F. Chang, L. Kennedy, and W. Hsu, \Columbia university's baseline detectors for 374 lscom semantic visual concepts," Columbia University, Tech. Rep., March 2007.

Cristianini, et al 2000 Cristianini, N. and Shawe-Taylor, J. (2000) An introduction to Support Vector Machines. Cambridge University Press. ISBN 0521780195

[Nigam et al. 1999] Kamal Nigam, Andrew Mc Callum, Sebestian ,and Tom. Text Classification from labeled and unlabeled document using EM. Machine learning. 1999.

[Tang et al. 2007] J-Tang, X-S Hua, G-J Qi, T Mei, X Wu. Anistropic manifold ranking for video annotation. In Proc. of the IEEE International Conference on Multimedia and Expo, 2007

Wu, et al. 2004 Yi Wu, Edward Y. Chang, Kevin Chen-Chuan Chang, and John R. Smith. Optimal multimodal fusion for multimedia data analysis. In Proceedings of the 12th annual ACM international conference on Multimedia, pages 572–579, 2004.

Tansley, 2000 Tansley, R. (2000). The multimedia thesaurus: Adding a semantic layer to multimedia information [doctoral thesis]. University of Southampton, UK.

Benitez, 2005 Benitez, A. (2005). Multimedia knowledge: Discovery, classification, browsing, and retrieval [doctoral thesis]. New York: Columbia University.

Bai, et al. 2007 Bai, Y., L. Di, A. Chen, Y. Liu, and Y. Wei, 2007. Towards a Geospatial Catalogue Federation Service, Photogrammetric Engineering & Remote Sensing, 73(6): 699-708.

Espinosa, et al. 2007 S. Espinosa Perald , A. Kaya , S. Melzer , R. Moller , M. Wessel, Towards a Media Interpretation Framework for the Semantic Web, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, p.374-380, November 02-05, 2007

Dasiopoulou, et al. 2008 Dasiopoulou, S., I.Kompatsiaris, and M.G.Strintzis (2008). Using fuzzy dls to enhance semantic image analysis. In SAMT’08: 3rd International Conference on Semantic and Digital Media Technologies.

Francois, et al. 2005 Francois, A., Nevatia, R., Hobbs, J., Bolles, R., Smith, J.: VERL: an ontology framework for representing and annotating video events. IEEE Multimedia 12(4), 76–86 (2005)

Hollink, et al. 2005 Hollink, L., Little, S., Hunter, J.: Evaluating the application of semantic inferencing rules to image annotation. In: Proc. of Int'l Conference on Knowledge Capture (2005)

Kang et al, 2004 Fang Kang, Rong Jin, Joyce Y. Chai, *"Regularizing translation models for better automatic image annotation"*, CIKM '04 Proceedings of the 13th ACM international conference on Information and knowledge management, ACM New York, NY, USA 2004

Kang et al, 2004 Fang Kang, Rong Jin, Joyce Y. Chai, *"Regularizing translation models for better automatic image annotation"*, CIKM '04 Proceedings of the 13th ACM international conference on Information and knowledge management, ACM New York, NY, USA 2004

[Bate. 1986] M. Bates, ―Subject Access in Online Catalogs: A Design Model, Journal of the American Society for Information Science, 11, 357 - 376, 1986.

[Furn et al. 1987] G. W. Furnas, T. K. Landauer, L. M. Gomez,, and S. T. Dumais, "The Vocabulary Problem in Human-System Communication", Communications of the ACM, Vol. 30, No. 11, November 1987,

Yang, et al (2006) C. Yang, M. Dong, & J. Hua (2006) Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 17–22.

Magalhaes et al. 2007 Joao Magalhaes, Stefan Ruger, 2007 "Semantic Multimedia Information Analysis for Retrieval Applications"